The Dissertation Committee for Natalia Ruiz Juri certifies that this is the approved version of the following dissertation:

Modeling the System-Level Impacts of Information Provision in Transportation Networks: an Adaptive System-Optimum Approach

Committee:

S. Travis Waller, Supervisor

Anant Balakrishnan

Carlos Caldas

Kara Kockelman

C. Michael Walton

Modeling the System-Level Impacts of Information Provision in Transportation Networks: an Adaptive System-Optimum Approach

by

Natalia Ruiz Juri, B.S.; M.S.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2009

To my family

Acknowledgments

I wish to express my gratitude to my advisor, Dr. Travis Waller, for his support, patience and encouragement throughout my PhD studies. Besides his guidance and advice in research-related matters, he gave me uncountable opportunities to actively participate in different aspects of the academic life. His trust and encouragement in this regard have greatly enriched my experience as a graduate student.

I also want to acknowledge the dedication and support of all the members of my dissertation committee: Dr. Kara Kockelman, who also guided my first research efforts in the United States, Dr. C. Michael Walton, Dr. Carlos Caldas, and Dr. Anant Balakrishnan, whose Supply Chain Management course, the first I ever took in optimization, had a decisive impact on my research interests.

I would like to thank my family, who has walked with me through every step of this process despite the distance that physically separates us, and my friends here, there, and wherever life has taken them.

My experience would not have been as enjoyable without the company of the graduate students in the transportation program, past and present, and it would have been considerably more challenging without the administrative assistance of Chandra Lownes and Libbie Toler, to whom I am deeply grateful.

And finally, I want to thank Dr. Avinash Unnikrishnan. With his unconditional support (and despite his insistence on cheering me up with "good jokes") he showed me that the true meaning of friendship is universal, and transcends culture and beliefs. This is, undoubtedly, one of the most valuable lessons I have learnt in this process.

NATALIA RUIZ JURI

The University of Texas at Austin
August 2009

Modeling the System-Level Impacts of Information Provision in Transportation Networks: an Adaptive System-Optimum Approach

Natalia Ruiz Juri, Ph.D.

The University of Texas at Austin, 2009

Supervisor: S. Travis Waller

Traffic information, now available through a number of different sources, is re-shaping the way planners, operators and users think about the transportation network. It provides a powerful tool to mitigate the negative impacts of uncertainty, and an invaluable resource to manage and operate the network in real-time. More information also invites to think about traditional transportation problems from a different perspective, searching for a better utilization of the improved knowledge of the network state.

This dissertation is concerned with modeling and evaluating the system-level impacts of providing information to network users, assuming that the data is utilized to guide an Adaptive System-Optimum (ASO) routing behavior. Within this context, it studies the optimal deployment of sensors for the support of ASO strategies, and it introduces a novel SO assignment approach, the Information-Based System Optimum (IBSO) assignment paradigm.

The proposed sensor deployment model explicitly captures the impact of sensors' location on the expected cost of ASO assignment strategies. Under such strategies, a-priori routing decisions may be adjusted based on real-time information.

The IBSO assignment paradigm leads to optimal flow patterns which take into account the ability of vehicles to collect information as they travel. The approach regards a subset of the system's assets as probes, which may face

higher expected costs than regular vehicles in the search for information. The collected data is utilized to adjust routing decisions in real time, improving the expected system performance. The proposed problem captures the system-level impact of adaptive route choices on stochastic networks.

The models developed in this work are rigorously formulated, and their properties analyzed to support the generation of specialized solution methodologies based on state-space partitioning and Tabu Search principles. Solution techniques are tested under a variety of scenarios, and implemented to the solution of several case studies. The magnitude and nature of the information impacts observed in this study illustrate problem characteristics with important theoretical, methodological and practical implications. The findings presented in this dissertation allow envisioning a number of practical applications which may promote a more efficient utilization of novel sensing and communication technologies, allowing the full realization of their potential.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Thanks to technological advances in the fields of remote sensing and communications, large amounts of transportation data have become available in the last 15 years (Sisiopiku [2000], Varshney [2003, 2005], Biesecker [2000]). Increasing proportions of such data are now accessible to network users in real-time, directly or as route recommendations from in-vehicle devices or Dynamic Message Signs.

The new available information invites to think about traditional transportation problems from a different perspective, searching for a more advantageous utilization of the improved real-time knowledge of the network state.

This dissertation introduces a novel system-optimum assignment approach which takes into account the ability of assets to collect information as they travel. The Information-Based System Optimum (IBSO) assignment paradigm captures the system-level impact of adaptive route choices on stochastic networks. The distinctive characteristic of the new approach is given by the utilization of some system assets as probes, which collect the information based on which the assignment of the remaining assets is adjusted.

Methodologically, this dissertation is concerned with modeling and evaluating the system level impacts of providing information to network users, assuming that the data is utilized to guide adaptive system-optimum routing strategies. Within this context it also studies the optimal deployment of sensors for the support of adaptive-system optimum decisions. Both contributions represent fundamentally new approaches to the collection and utilization of information in transportation networks.

Existing literature on adaptive routing typically considers that the information based on which routing decisions are adjusted is either limited to the costs individually experienced by each asset, or exogenously provided from arbitrary sources. The first case gives rise to models of shortest path with recourse (e.g. Polychronopoulos and Tsitsiklis [1996], Miller-Hooks and Mahmassani [2000], Waller and Ziliaskopoulos [2002]), while the second approach underlies studies concerned with in-vehicle route-guidance provision (e.g. Papageorgiou and Messmer [1991], Boyce et al. [1995], Friesz et al. [1989]). Under an IBSO assignment paradigm information is both, endogenously generated, and systemic. This leads to framework which captures the system-level impacts of information, and at the same time models its collection within a system optimum context.

The optimal sensor deployment approach proposed in this dissertation (Section 4) considers exogenous information sources, but it explicitly studies the impact of the location of the source on the effectiveness of the route guidance strategies. The model identifies the optimal location of a fixed number of static sensors in a network with stochastic arc costs, in such way that the expected cost faced by a set of optimally routed assets is minimized. The information provided by the sensors translates into a set of perceived states, based on which adaptive routing decisions are made. Section 1.1.3 exemplifies a possible scenario for the application of the proposed approach.

Under the IBSO assignment paradigm presented in Section 6, a subset of system assets are designated as probes, and used to collect the information used to adjust the assignment decisions for the remaining assets. The selection of the paths followed by the probes takes into account the value of the information collected along them in addition to the corresponding expected cost. As a consequence, assets utilized as probes may face higher expected costs than regular system assets, which are optimally routed under every possible state revealed by the probes. A multitude of problem variations are possible based on the assumptions regarding the characteristics of the network, the information distribution scheme, and the timing of the deployment of probes and regular assets. Section 1.1 presents a taxonomy to classify the variations, along with the set of assumptions governing the problems studied in this dissertation, and Sections 1.1.2 and 1.1.1 illustrate possible applications.

The problems addressed in this work are formulated as stochastic programs (Sections 4.2 and 6.3), which allows explicitly modeling uncertainty, information provision and utilization. Problem solutions have a deterministic component, given by the sensor/probe deployment strategies, and an adaptive element. The later is given by the strategies which define the routes to be followed by the non-probe system assets under each possible scenario, typically represented using hyperpaths (Nguyen and Pallottino [1989]).

The solution methodology is based on the fact that, in virtue of the assumptions presented in Section 2.7, the models may be solved by enumerating all feasible sensor/probe deployment strategies, and computing the corresponding expected costs under information. Such approach poses two main challenges: the large number of perceived states which need to be considered during the evaluation of a feasible deployment strategy, and the existence of a combinatorial number of strategies. The proposed solution technique deals with the first issue using state-partitioning principles, while the combinatorial problem is addressed heuristically, by implementing an adaptive memory Tabu search procedure. Both methodologies were tailored to account for the characteristics of the problem under study, and their performance tested under several scenarios. The final methods constitute an interesting framework for the study of similar problems.

The models proposed by this work are implemented to the study of several numerical examples. The analysis of the results from a quantitative and qualitative perspective, presented in Sections 5.3 and 6.6, illustrate interesting problem characteristics with important theoretical and methodological implications.

The presented models are not designed to solve any specific application, but a number of possible implementations are possible for the different variations summarized in Chapter 2. The following section exemplifies some potential practical implementations, while Section 1.2 summarizes the goals, objectives and contribution of this dissertation.

1.1 Motivating examples

This section presents some simple examples of scenarios under which the strategies and methodologies proposed by this work would be advantageous. These cases motivate the present research, by illustrating the potential benefits of peer-based dynamic and targeted information collection and utilization, and also to facilitate the understanding of some of the problem properties and variations defined on later chapters.

1.1.1 Evacuation of damaged areas after a natural disaster

Consider the evacuation of an area affected by a natural disaster, such as flooding. Assume that the network presented in Figure 1.1 represents the corresponding road system, which has been damaged by the event. Suppose that, based on their previous experience, emergency managers are able to estimate the probability of different damage levels on the network links, which define the corresponding link traversal costs. Assume that the objective of the rescue operation is to transport people affected by the natural disaster, currently concentrated in node C, to a safer location in node G. The evacuation needs to be finalized in the shortest possible time, given the danger of a sequel, and it is accomplished using 4 vehicles, which are enough to transport all the evacuees. A naive routing strategy would assign all the assets to paths CDFG or CDEG, which exhibit the least expected cost, equal to 7 time units.

Nevertheless, it is possible to improve upon this strategy by making use of wireless communications capabilities and a staggered decision making process.

For example, two of the assets may be assigned to paths CDFG and CDEG at the onset, and then routes for remaining vehicles decided based on their findings. Under this strategy, the expected cost to be paid by assets routed on the second stage becomes 5.5 units, and the expected total system cost is reduced from 28 to 25 time units. Taking into account that under the proposed deployment strategy assets assigned during the second stage must wait until the latest of the first two vehicles reaches destination, the expected arrival time of the last vehicle is 12 time units.

Notice that this strategy involves assigning probes to paths exhibiting higher expected costs than the minimum expected cost path. However, the possible gains derived from the information found by the probes outweigh the additional cost, in terms of expected travel time, necessary to pay for such data.

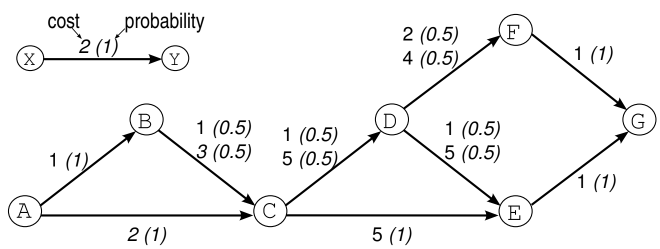

Figure 1.1: Example network

1.1.2 Deployment of military assets in a hostile environment

Consider a set of military vehicles consisting of tanks and ambulances, which need to be deployed through a hostile territory. Let the network in Figure 1.1 represent the set of paths traversing the area, and assume that the random link costs are a function of its length and condition, as well as of the risk of being attacked while traveling on the link. Also consider that the tanks, more resilient to enemy attack and adverse road conditions, are used as probes and deployed during a first stage. The ambulances are vulnerable, and therefore routed during the latest stage.

Let the objective of the assignment be the displacement of the assets from their current location in node C, to a new settlement in node D. Even though it is desirable to accomplish the relocation in the shortest possible time, a major concern is to ensure that the time spent on the road by ambulances is minimized, in order to reduce their exposure, and the risk to the transported patients. Taking such goals into account, the deployment strategy presented in the previous example may be improved upon, by exploiting the fact that some paths have links in common. In view of the later, greater information benefits for the vehicles deployed in later stages can be achieved if each probe vehicle waits for the previous one to reach its destination before entering the network. Such strategy allows probes to utilize the information gathered by others. The expected cost faced by the ambulances under this strategy is reduced from 5.5 to 5.25 time units, at the cost of a higher arrival time for the last unit (17.4). The corresponding system expected cost becomes 24 time units.

1.1.3 Optimal deployment of sensors for decision-making support

Consider that the network in Figure 1.1 represents the transportation system of a region susceptible to flooding. Assume that link travel times are a direct function of the water level on a link, which follows a discrete probability distribution. Suppose that city planners are able to deploy a fixed number of sensors able to measure the actual water level on a link. The objective underlying the sensor placement is to guarantee the fastest possible deployment of help to flood victims, which usually seek for refuge in shelters and hospitals located in node D. Emergency vehicles typically depart from the fire station located at node A. Assuming that there is only one sensor available, the clear choice for its placement is link BC. Notice that locating the sensor on link CD would provide a more accurate estimate of average level of water in the system, which is desirable from a monitoring perspective. However, such information is worthless for the specific objective under consideration, because it has no impact on the decision making process, which is certain to involve link CD in the assignment. By using the information provided by the sensor, the expected total system cost drops from 4.5 to 4 time units.

1.2 Goals and objectives

This dissertation aims to contribute to a better understanding of the system-level effects of providing traffic information to network users, thus fostering the development of methodologies capable of strategically exploiting such impacts to benefit the system. Its goal is to introduce new routing paradigms benefiting from the increasing availability of real-time data, ultimately leading to a more efficient utilization of available resources, and a full realization of the potential benefits of novel technologies. Specific objectives include:

- Study existing approaches to evaluating the system-level impacts of information provision to network users.
- Propose a novel sensor deployment criteria based on the impacts of information on adaptive routing decisions.
- Define a new assignment paradigm capable of benefiting from emerging sources of real-time traffic data.
- Introduce a framework for the study of the novel approach, proposing a taxonomy to classify problem variations.
- Formulate a mathematical model reflecting the proposed concepts.
- Analyze properties and distinctive characteristics of the new model.
- Develop and implement specialized solution methodologies.
- Identify possible extensions and applications of the proposed framework based on the analysis of problem properties and numerical results.

This work contributes to the literature on the evaluation of information impacts, by presenting methodologies able to capture and measure the effects of providing information to network users. It also advances the research concerned with the optimal deployment of sensors, by proposing models which explicitly consider the impact of sensor placement on the performance of adaptive strategies based on the information they provide.

The proposed Information-Based System Optimum assignment paradigm is a fundamentally new approach that allows exploitation of traffic information from wholly new perspectives. The analysis of numerical results and theoretical problem properties suggests that the paradigm has the potential to improve system performance. The models proposed in this work constitute an initial step towards enhancing information collection and utilization strategies. Based on the findings presented here, a number of applications may be envisioned, for which efficient solution would ensure more efficient utilization of current and future sensing and communication technologies, fostering the full realization of their potential benefits.

Chapter 2

Conceptual Framework, Problem Variations and Assumptions

This dissertation is concerned with modeling and evaluating the system level impacts of providing information to network users, assuming that the data is utilized to guide adaptive system-optimum routing strategies. Within this context it studies both, the optimal deployment of sensors for the support of adaptive-system optimum decisions, and the design of optimal assignment strategies which consider the collection of information by system assets.

Adaptive decisions are such that they may be modified based on available information, whether it originates from static sensors or it is provided by other assets in the system. The system-optimum approach implies the implementation of a routing criterion which minimizes the total cost faced by the system, at the price of allowing some assets to face higher costs than others (Sheffi [1985]).

In the context of this work, the system costs are defined by the summation of the expected costs paid by individual assets. The use of expectation is motivated by the stochastic nature of the problem, in virtue of which the actual cost at a link is learnt only after making the decision to utilize it. Even when the expected cost of a path is minimum, the conditions experienced by the assets traversing it may be worse than the realized costs in other paths. The later explains the advantage of using probes to verify the state at one or more routes before making the assignment decision for the remaining vehicles, which is the concept underlying the Information Based System Optimum (IBSO) assignment paradigm.

Section 6.2 explains some unique characteristics of the IBSO assignment approach, and contrasts it to traditional system-optimum assignment. The problem involves deciding the routes to be followed by a set of assets, including probes, which travel between given origins and destinations. Routing decisions are made a-priori, but they may be staggered in order to allow for the utilization of the information retrieved by the probes. Given the stochastic nature of the problem, solutions are given in the form of strategies, rather than as a specific set of routes. Strategies describe the optimal assignment under each possible scenario, and may be represented using hyperpaths.

The approach taken in this dissertation to study the optimal deployment of sensors assumes that the collected data is used to optimize system-optimum routing decisions, and explicitly models the impact of the sensor location on the resulting system performance.

The general problem description provided in this section makes it clear that there is a multitude of possible variations depending on the assumptions regarding various problem parameters, which ultimately define what information is available and which assets may benefit from it. The following sections describe these parameters, identifying axes along which the problem variations may be compared, and defining the scope of the present work.

2.1 System assets and routing strategies

One of the main features of the IBSO assignment problem is the consideration of two types of assets. An arbitrary set of assets is utilized as **probes**, also denotes equipped assets, which are capable of collecting and communicating information about the system state as they travel. The data revealed by probes may be used to adjust the assignment decisions concerning other vehicles.

The non-probe assets, denoted regular vehicles or **non-equipped assets**, experience link cost realizations on their trips, but their findings do not have an impact on routing choices.

All assets are assigned to routes connecting an origin-destination pair seeking to optimize a common system-level objective. Given the capability of probe vehicles to resolve the system uncertainty by collecting information, may be routed on different (and more expensive) paths than the regular assets.

In this dissertation all routes are assumed to be selected **a-priori**, i.e. before assets leave the origin. The adaptive component is given by the adjustment of route decisions based on the available information, which is possible assuming a serial deployment process described in section 2.4. Possible problem variations include allowing the re-optimization of routes at intermediate nodes based on information retrieved by probes after the departure of regular assets (**system-level-information-based recourse**). Another approach is to allow regular assets to re-route themselves based on the costs they experience, following an online shortest path approach (e.g.Waller et al. [2001]). If flow dependant link costs are considered, the later would not be consistent with a system optimum approach, but a system-optimum with recourse paradigm (Unnikrishnan [2008]) may be use to centrally define assets routes.

2.2 Network characteristics

System assets are assigned into a directed, uncapacitated network with stochastic link cost functions. Link costs follow discrete probability distributions, defined by a finite number of states, which are uncorrelated across links. Additionally, the considered network is static, which implies that link cost probability distributions are constant in time. It is assumed that probe vehicles learn the cost of a link upon traversing it, and that such cost becomes deterministic for the purpose of assigning the remaining assets, as well as invariant. The later means that additional probes visiting the same link would experience the same cost. Costs functions are assumed to

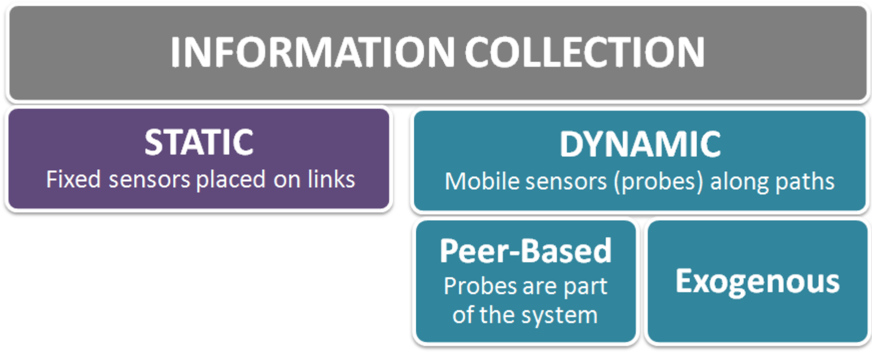

Figure 2.1: Classification of information collection strategies

be independent of the corresponding flow and, without loss of generality, integer-valued (Ahuja et al. [1993]).

Some network nodes act as origins and/or destinations, which are the entry and exit points of the assets into the system. Problem formulations presented in this proposal consider a single origin and destination pair.

2.3 Information collection and distribution

For most of the problems proposed in this dissertation, information is dynamically collected, in the sense that it is retrieved by mobile assets traveling through the network. Chapter 4 presents the only problem version involving static information collection, where the same is redeemed by sensors placed on fixed network links. In both cases, links from which information is collected are considered to be "measured". Figure 2.1 depicts possible assumptions regarding information collection. Information is collected by system assets, even though further extensions may consider the utilization of exogenous agents to generate network-state data.

In the context of this problem, the direct impact of measuring a link is given by a change on the corresponding cost, from its expected value to the value observed under each possible realization in its probability distribution. The combination of the realizations observed at every measured link originates a "perceived" network state, based on which adaptive routing decisions may be performed.

Equipped and non-equipped assets learn the realization of an arc cost upon traversing it, but the uncertainty regarding a link cost from a system perspective is only resolved when a link is traversed by a probe, and upon the same arrives to an **information retrieval** node. These are the only nodes from which probes can make information available to the system. Similarly, **information distribution** nodes are defined as those nodes at which regular assets may be re-routed. The approaches proposed in this dissertation consider single information retrieval and distribution nodes, given by the origin and destination of the assets route. This represents the a-priori routing strategy described in 2.1. Problem instances allowing recourse actions require the definition of several information distribution nodes.

2.4 Deployment strategies

Deployment strategies define the structure of the decision making process. Two main approaches are possible depending on whether probes remain in the system after regular assets are deployed. **Parallel deployment strategies** (Figure 2.2) assume that both types of assets are assigned into the network simultaneously. Under serial strategies non-equipped assets enter the network only after the probe vehicles have reached their destination. A serial approach has the potential to result in lower system expected costs, given that the assignment decision for regular assets is supported by all the information which the probes may retrieve. However, if the total time spanned by the deployment is a concern, parallel strategies may be more appropriate. Notice that, while serial deployment is compatible with a-priori routing schemes, some of the alternatives within parallel deployment are meaningless if system-level-information-based recourse (Section 2.1) is not allowed, at least for the non-equipped assets.

Serial deployment strategies can be further categorized as simultaneous or sequential based on the ability of probes to take advantage of the findings of other equipped assets. Under a simultaneous assignment paradigm, all

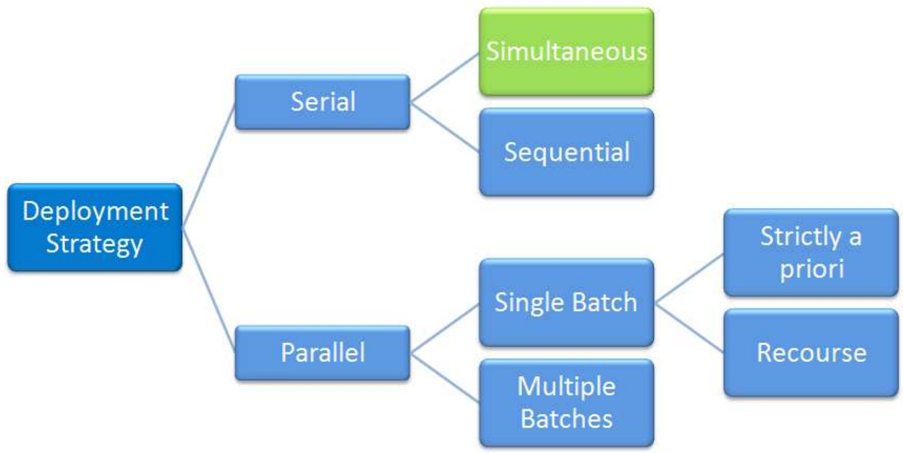

Figure 2.2: Deployment strategies

equipped assets enter the system at the same time, and collect information independently of the findings of other probes. The **sequential probe assignment** approach assumes the equipped assets are released into the system in batches, and the routes of probes in later batches are decided based upon the information collected at earlier stages. The sequential assignment option may lead to a more efficient use of the information at the cost of prolonging the total deployment time. Additionally, sequential strategies entail the introduction of a temporal dimension, in order to ensure consistency between the time at which batches are deployed into the system (**release time**) and the corresponding availability of information. If link costs do not represent travel times, additional variables are required to maintain such consistency.

Under parallel deployment strategies, it is possible to release all the assets into the system at once, or to accomplish the deployment in batches consisting of non-equipped assets and probes. The proportion of each type of asset per batch may be fixed exogenously, or considered a problem variable. An additional parameter to define in this context is the location and number of information retrieval and distribution nodes (Section 2.3). The assumption of unique retrieval and distribution points at the origin and destination of each route may be excessively restrictive. Depending on the release time of the different batches, it may actually prevent some batches from taking any advantage of the information collected by other probes. A more flexible approach is to let all the nodes be information retrieval nodes, allowing every batch to benefit from all the information acquired before its release time. In addition to this, cases involving a single batch consisting of all the assets in the system demand for extra information distribution nodes, and entail allowing for system-level-information-based recourse.

2.5 Objective function

For the purpose of this dissertation, the system objective is to minimize the **summation of the expected cost paid by each asset**. Notice that this assumption minimizes the time spent by the assets in the network rather than the expected duration of the deployment operation. The optimal solution to problems with this objective may tolerate considerably longer routes for probe vehicles, which is compensated by the availability of more valuable information concerning the system state at the moment of assigning the remaining assets. This objective function is reasonable for problems such as the ones presented Section 1.1.2, in which the main concern is the time spent en-route. For problem instances such as the one presented in Section 1.1.1, in which the total time involved in the deployment process is of importance, it may be desirable to re-define the corresponding objective function and incorporating a temporal dimension. If the link costs are not expressed in time units, it may demand for the definition of new variables, and lead to multi-objective problems. Otherwise, the objective function may be adjusted by adding the expected arrival time of the last probe to the cost summation defined earlier.

2.6 Decision variables

For the IBSO assignment problem, the decision variables include the route to be followed by the probes and the assignments strategy for regular assets under each possible perceived scenario. Notice that in virtue of the assumption of flow-independent and uncapacitated link costs, all system assets may be assigned to the same path, and therefore the solution involves a single path per state. The set of the different solutions under each network state is denoted hyperpath (Nguyen and Pallottino [1989]). For the optimal sensor deployment problem, the decision variables are given by the set of links to be measured, and the corresponding hyperpaths. Future extensions may incorporate additional degrees of freedom, including an endogenous selection of the number of assets to be used as probes, or the release time of each batch in parallel deployment cases under a sequential release strategy.

2.7 Summary

This section described the features of adaptive system optimum routing strategies, and presented the possible assumptions defining an instance of the IBSO assignment problem. The characteristics presented in this chapter constitute the axes along which problem variations may be defined and compared. The number of variants resulting from combining different assumptions regarding problem parameters is certainly very large. In order to bound the scope of this work, the analyzed version will differ only on the deployment and routing strategies, as categorized in Figure 2.2. For the remaining parameters, Table 2.1 presents the assumptions which will be held throughout this work, unless otherwise noted.

Table 2.1: Assumptions

Network	Directed, static, uncapacitated, single OD pair
Link costs	Integer, non-negative, flow independent
Cost probability distribution	Discrete. Realization learnt after use by probes.
Information retrieval node	Destination
Information distribution node	Origin

Chapter 3

Literature Review: Evaluating the Impacts of Traffic Information Provision on the Performance of Transportation Networks

Thanks to technological advances in the fields of remote sensing and communications, large amounts of transportation data have become available in the last 15 years (Sisiopiku [2000], Varshney [2003, 2005], Biesecker [2000]). This data typically includes vehicle counts and speeds at specific locations, as well as point-to-point travel time measurements from "probe" vehicles equipped with wireless devices such as toll-tags, GPS, and even cellular phones. The existing intelligent transportation system infrastructure allows collecting, processing, and distributing information from some of these sources in almost real time.

The generated information is typically made available to both, network managers and drivers. For network operators, the ability to closely monitor the system performance has an enormous value. The collected data contributes to a better understanding of the behavior of transportation networks, and provides a means to develop more efficient congestion management and network operation strategies.

Evaluating the effects of providing information to drivers is considerably more complex. The availability of real-time data is likely to affect the route and departure time choice of system users, especially under congested or atypical situations. However, the aggregate impact of individual choices on the system behavior is not easy to predict, particularly given the difficulty of assessing the reaction of drivers to information.

This literature review presents existing approaches to model the utilization of information by drivers in transportation networks (Section 3.1) and the corresponding system-level impacts (3.2). These models provide useful tools to evaluate information collection and distribution strategies (Section 3.3), which is critical in view of the large costs that such tasks may involve. They may also serve as the basis to analyze novel approaches to the utilization of information in transportation problems, and therefore understanding their strengths, limitations and corresponding implementation challenges is of the utmost importance.

3.1 Modeling the utilization of information

This section focuses on models which are able to represent the adaptive behavior of drivers in the face of information. The methodologies studied here typically assume that all drivers optimally use the available information to improve their travel cost (or expected travel cost). In reality, drivers' reaction to information is a more complex process which depends on their preferences, perceptions, past experience, and attitude, among others. The study of such behavior escapes the scope of this work, but it is an active field of research, typically accomplished via interactive simulation experiments and surveys (Mahmassani and Chen [1991], Polak and Jones [1993], Adler and McNally [1994]Mahmassani and Tong [1986], Koustopoulos et al. [1993]). Another assumption common to most of the models discussed in this section is that link costs are random, reflecting the uncertainty capacity which characterizes transportation network links (Unnikrishnan [2008]). In this context, information is represented as the total or partial resolution of uncertainty accrued by learning the cost realization on one or more links.

The methodologies analyzed in this section may be classified according to multiple characteristics. Figure 3.1 presents the approach selected for this study, which is based on the type of information provision that the models may be used to represent. It is important to notice that many of the reviewed models were not developed to meet the requirements of a specific technology, but to capture realistic behaviors which were beyond the capability of past approaches. The proposed classification scheme is designed to fit the framework of this dissertation, and the corresponding categories are intended to describe possible model application within the context of this study. The proposed classification is not meant to define the entire range of possible implementations of the discussed research efforts.

For the purpose of this study, **self-collected information** corresponds to the information that vehicles learn as they travel through the network, either by reaching a node or by traversing a link. It may include information displayed on dynamic message signs, as long as such data is concerned only with the state of adjacent links. The models included in this category are mostly variations of the traditional shortest path problem, born from the need to explicitly model link cost uncertainty, adaptive drivers behavior, and dynamic network properties.

System-level information may originate at any location in the network, and is typically collected and distributed using the Intelligent Transportation System (ITS) infrastructure, even though novel approaches explore the feasibility of propagating system-level data through vehicle-to-vehicle communications (Section 3.3). It may be accessed through the Internet, or by means of an in-vehicle navigation device. Most of the models in this section were developed to study the provision of dynamic route-guidance to drivers.

3.1.1 Models based on self-collected information

The models in this section explicitly account for the uncertain nature of the link costs, and capture the adaptive behavior of drivers. On stochastic

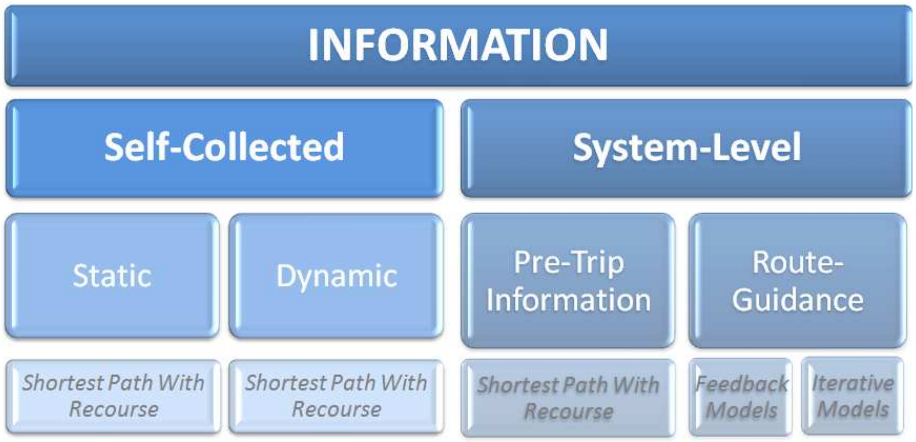

Figure 3.1: Approaches to modeling and evaluating individual-level information impacts

networks one may not find a path which is the shortest under all possible cost realizations. A common practice is to define optimality based on paths expected cost, regarding the least expected cost (LEC) path as the "shortest path", which can be found using deterministic shortest path algorithms when the link cost functions are linear (Eiger et al. [1985]). For many transportation applications, this is a very sensible choice which reflects driver's decision-making process under uncertainty (Von Neumann and Morgenstern [1944]). However, for some implementations, particularly those involving short-term decision making, modeling uncertainty becomes crucial.

Online optimization relaxes the assumption that the entire optimal path must be chosen before arc costs realizations are learnt. Drivers select an a-priory route, but are allowed to adapt their decision as they travel based on the cost realizations they observe. Furthermore, the formulations are such that the a-priori route selection takes into account the information which may become available on the selected path, and the corresponding alternatives. The changes to the a-priori route selection are usually referred to as "recourse", and are typically assumed to be governed by a cost minimizing process.

The optimal solution to this type of models is given in the form of a hyperpath, which is an acyclic graph representing the optimal route given different link (nodes) costs realizations (Nguyen and Pallottino [1989]). The expected cost of following an optimal hyperpaths is a lower bound on the expected cost of any a-priori path. The major challenge posed by the solution of these problems is that, in the general case, they may involve the enumeration of all existing hyperpaths, which is very large and grows exponentially with the problem size. However, some efficient methodologies have been developed for specific link correlation structures and network topologies.

Models may differ in a multitude assumptions, including the point at which the driver learns a link's cost realization (origin or destination node), the permanency of such cost realization (it may be different each time a node is visited or become deterministic after the first pass), and the type of correlation among link cost probability distribution functions (independent, arbitrary correlations, or specific structures). Some models also account for dynamic network properties, assigning different probability distribution functions depending on the arrival time at a node. Gao and Chabini [2006] present the first somewhat unifying framework to analyze online shortest path problems on networks with discrete and non negative arc costs, and proposes very general optimality conditions.

Hall [1986] introduces a dynamic programming algorithm to find the shortest hyperpaths on a network with stochastic arc costs on which drivers are allowed recourse actions based on their experienced costs. His methodology assumes that the least expected cost path is acyclic, and it may require evaluating all possible paths.

Psaraftis and Tsitsiklis [1993] study the problem on acyclic networks which arcs exhibit a Markovian temporal dependency (i.e. the arc cost at state t-1 is a function of the arc cost realization at t) but are uncorrelated space-wise. In the proposed models, waiting at nodes is allowed. The authors present three polynomial algorithms to find optimal adaptive routes, based on dynamic programming principles.

Bander and White [2002] study a problem variation including path terminal costs, which depend on the corresponding arrival time. The methodology assumes discrete arc cost probability distributions and positive arc costs, and may be solved heuristically.

Polychronopoulos and Tsitsiklis [1996] consider the online shortest path problem on networks with discrete arc cost probability distributions under two possible conditions: independent and correlated arcs cost. For the correlated case, they define a state set, which contains all the possible network realizations. This work introduces a dynamic programming algorithm which recursively reduces the cardinality of the information set. The rationale behind such approach is that drivers may "discard" some of the possible network realizations as they acquire new information. The authors propose exponential solution methodologies and bounded heuristic approaches.

Waller and Ziliaskopoulos [2002] examines a similar problem, but assumes arc costs probability distributions with limited space dependencies, which lead to a polynomially solvable problem. The authors consider that each visit to a node results in a new random trial, and therefore the shortest path could include cycles (Andreatta and Romeo [1988]). As a result, the proposed label correcting algorithm may theoretically lead to infinite cycling. However, this contribution proves that there is a bound to the maximum improvement (in terms of expected cost) that can be derived from cycling, and provides an heuristic bound on the algorithm performance based on the desired level of precision in the solution.

Miller-Hooks and Mahmassani [2000] present an algorithm to compute least-expected cost paths in **stochastic dynamic networks**, on which the cost functions are not only random but also time dependant. The label-correcting methodology provides an exact problem solution in networks with independent arc costs, assuming that no waiting is allowed at nodes. The algorithm is exponential in theory, but it is showed to perform considerably better in practice. The proposed implementation is possible thanks to the additive nature of the expected costs, guaranteed by the characteristics imposed on the probability distribution functions. De Leone and Pretolani [1998] and Pretolani [1998] analyze the same problem, developing solution methodologies based on auction algorithms and time-expanded networks, respectively. Such methodologies work in linear time with respect to the corresponding network size, which may however be very large.

Models accounting for the adaptive behavior of drivers when presented with information are the building blocks for formulations able to represent the aggregated behavior of the system under the provision of information. Existing methodologies are insightful and flexible, and they may be adjusted to model they availability of new sources of information.

3.1.2 Models based on system-level information

New technologies, such as GPS and in-vehicle guidance systems, allow assuming that a large set of system-level information is available at different points during the trip (even continuously), and that optimal route choices may be made based on it. Most of the online routing models accounting for centralized information were motivated by the emergence of in-vehicle guidance systems capable of accounting for real-time traffic information. As a result, they focus on generating route recommendations which are provided to drivers instead of the corresponding raw data.

Some of the models presented in this section are designed to provide only pre-departure information, which may have an effect on both, route choice and departure time selection. Providing pre-trip information also presents a better opportunity to seek for a system performance balancing system optimum and user equilibrium objectives. Jahn et al. [2005] propose a system-optimum approach with user constraints towards such end. The methodology generates optimal routing strategies which improve upon the solution of a user equilibrium problem (Sheffi [1985]) while limiting the magnitude of the cost differences that characterize system optimum assignment strategies. Notice that the approaches presented here do not necessarily model uncertainty through a random distribution of link costs. Some of the presented models approach the problem from a dynamic traffic assignment perspective (discussed in Section 6.1), in virtue of which travel time realizations are the result of a simulation process.

3.1.2.1 Pre-trip information provision

Not many route-guidance implementations are limited to **pre-trip information provision**. The algorithm provided by Miller-Hooks and Mahmassani [2000] may be applied for such purpose, given that it identifies the least expected cost path on a stochastic and dynamic network on which arc travel times are represented by discrete probability distributions. She implements a modified label correcting algorithm to identify all the paths with a positive probability of being the shortest from all origins to a single destination. These non dominated, or Pareto-optimal, paths are then compared to select the optimal route. Since all the paths in a network could be Pareto-optimal, this algorithm has a non-polynomial worst case complexity. However, it was found to perform more efficiently experimentally.

Sen et al. [2001] propose a slightly more complex approach which considers not only the expected cost of a path, but its variability, as given by the corresponding variance. They propose a mean-variance model, formulated as a convex quadratic problem which can be solved efficiently using interior point methods. The authors test the proposed models in toy networks, and discuss the availability of real data for practical implementations. Their findings suggest that the mean-variance approach is particularly appropriated for grid networks, on which turning movements may have a considerable impact on the reliability of travel times.

Chen et al. [2005] propose a route guidance methodology which combines reliable a-priori path selection with dynamic route guidance based on real-time data. The two-folded approach is intended to reduce the computational burden involved in en-route shortest path re-computations. Users are provided with a set of alternative a-priori paths instead of a single route, seeking to limit the effects of concentration (Lee [1994]). The path set is generated based on historic data implementing a criterion which reduces the probability of joint path failure, thus minimizing the number of re-optimization instances. The later are accomplished implementing the A* algorithm (P.Hart et al. [1968], Klunder and Post [2006]), which benefits from the a priori path information and performs very efficiently.

The methodologies in this section are useful to estimate the general changes in the route choice patterns that the provision of information may introduce. However, accounting for the adaptive behavior of drivers when presented with information is important in the search for further realism, and crucial for the evaluation of some congestion management and traffic operation strategies.

3.1.2.2 Dynamic route guidance

Dynamic route guidance strategies have been studied from many different perspectives, which Pavlis and Papageorgiou [1999] classify into iterative strategies and feedback strategies. The fist group includes those models which consider the impact of the guidance action on the experienced travel times and perform iterations until "equilibrium" flows are found, according to some pre-specified control objective. These methodologies, which often seek to attain dynamic traffic assignment optimality conditions, are typically more demanding from a computational perspective (Papageorgiou [1990], Papageorgiou and Messmer [1991], Charbonnier et al. [1991], Messmer and Papageorgiou [1994], Mahamassani and Peeta [1994].). Feedback strategies provide recommendations based on instantaneous traffic conditions, disregarding traffic dynamics and evolution. Strategies in both categories may be approached as a control theory problem, given that ultimately they can be represented by the corresponding split ratios at decision nodes (Papageorgiou and Messmer [1991], Boyce et al. [1995], Friesz et al. [1989]). Schmitt and Jula [2006] suggest additional classifications of these methodologies, distinguishing between centralized and decentralized approaches, and deterministic and stochastic models, among others.

Pavlis and Papageorgiou [1999] compare the performance of feedback and iterative strategies utilizing a simple traffic model. Their experience suggests that, even though iterative strategies are a more accurate representation of reality, feedback strategies may produce comparable results under specific conditions typically present in mesh networks. This is appealing because feedback strategies are considerably easier to implement efficiently as a set of decentralized control laws.

An example of an iterative strategy approach is the work by Boyce et al. [1995], who apply optimal control theory to model the optimality conditions of the dynamic user-optimum assignment problem. They arrive at a discrete non linear program formulation, which may be solved at each node based on instantaneous information. A variation of Frank-Wolfe's (Frank and Wolfe [1956]) algorithm for time-expanded networks is implemented towards this end. Kaufman et al. [1998] approach a similar problem from the system optimum perspective, and solve it as a mixed integer program on a time expanded network.

Feedback strategies are often reduced to the solution of a re-optimization problem at each decision node. Appendix B discusses some efficient algorithms designed to re-optimize shortest paths after link costs are updated. Fu [2001] presents an approach to re optimize the online shortest path, which they implement using dynamic programming. The label correcting algorithm they present is shown to perform no worse than the version typically used to compute shortest paths.

The study of dynamic route guidance strategies is an active area of research, given its practical applications and the role that the corresponding models play in understanding the aggregated behavior of networks under uncertainty.

3.2 Evaluating the system-level impacts of information utilization

Preliminary attempts to evaluate the effects of real-time information provision on transportation networks focused on analyzing the potential benefits and disadvantages of such strategy from a somewhat qualitative perspective.

Fujii and Kitamura [2000] conduct a survey to analyze the impact of traffic information on the **decision making process of drivers** using a specific freeway, subject to closures. They study whether or not driver's estimation of route travel time, typically based on their previous experience, may be altered by providing information. Their results do not exhibit any statistically significant trend. However, Srinivasan and Mahamassani [2002]'s findings suggest that real-time data provision has a strong impact on drivers route and departure time choice under changing traffic conditions using an interactive traffic simulator. Kitamura and Nakayama [2007] analyze the same subject from a theoretical perspective, modeling it as a minority game. Their findings suggest that the provision of information, even when accurate, can not affect the system behavior in the long term. According to the model proposed in this work, drivers always re-organize themselves in identical ways, reaching the same equilibrium point regardless of the availability of predictive information.

Yoshii and Kuwahara [2000] analyze the possibility introducing **negative system-level impacts,** such as increased delays or travel times, by providing information along a major arterial street. His results, which are consistent with previous findings by Moritsu [1991], indicate that such impacts are possible, particularly in underutilized networks. Similarly, research analyzing the provision of in-vehicle guidance information (Mahmassani and Chen [1991], Oh and Jayakrishnan [2002], Watling and Van Vuren [1993], Arnott [1991]) suggests that the system performance may deteriorate for market-penetration levels higher than 20%-40% are achieved, due to overreaction and concentration effects. Ben-Akiva et al. [1991] classify and analyze both, positive and negative impacts of information, and proposes an analytical demand and route choice model for their analysis.

The advent of route-guidance systems motivated the development and implementation of complex models, capable of providing rigorous performance measures. These models (Figure 3.2) have been used to understand the conditions under which adaptive route guidance is beneficial, and to identify the potential negative impacts of information provision on the system performance.

The models used to evaluate the impacts of information in transportation networks fall within two main categories: Simulation methodologies and equilibrium models.

Simulation models are appealing given their flexibility. They allow the incorporation of complex assumptions regarding driver's reaction to

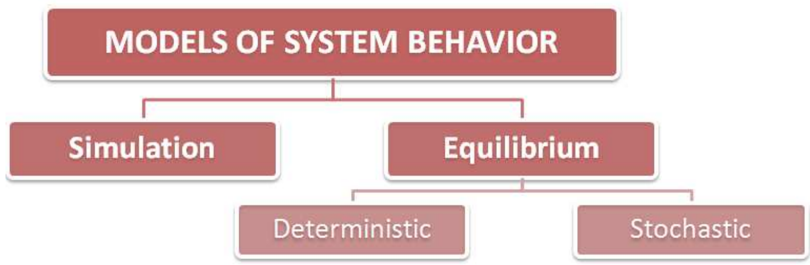

Figure 3.2: Approaches to modeling and evaluating system-level information impacts

information and information distribution schemes. They are adequate to capture "reactive" behavior, reflecting the changes in local conditions which result from the provision of information under a specific incident, or as a response to a traffic management technique. However, simulation is not capable of modeling the system-level changes introduced by the consistent provision of information to drivers, which are likely to involve route changes.

Equilibrium approaches aim to provide analytical expressions describing the aggregated behavior of a system in which drivers behave according to the models presented in 3.1.2. This approach provides an ideal framework for the study of system level information impacts, because it is able to capture the reaction of drivers to both, information and the behavior of other system users. However, the existence and uniqueness of an equilibrium solution under the adaptive behavior described earlier is not necessarily guaranteed under general conditions. Unnikrishnan [2008] uses a variational inequality approach to prove the existence and uniqueness of an equilibrium solution in stochastic networks with random capacities, in which drivers learn the cost probability distribution of a link when they reach its origin node. Marcotte et al. [2004] extends an equilibrium formulation originally developed for transit networks with hard capacities to the case of traffic assignment on static networks. Using a shortest hyper path approach and variational inequalities, this work proves the existence of an equilibrium solution and provides solution algorithms. Hamdouch et al. [2004] present the dynamic version of the problem. Gao [2005] suggests a policy-based dynamic traffic equilibrium approach, but does not provide analytical expressions.

Even though equilibrium with recourse models have not been deployed in large networks, they provide an appealing, theoretically sound approach to the assignment problem under uncertainty, which may be used to better understand the behavior of networks under information.

Dynamic Traffic Assignment Models incorporate the time dimension within the user-equilibrium (or system optimum) assignment paradigm, seeking for a more realistic description of the day-to-day network behavior (Peeta and Ziliaskopoulos [2002], Merchant and Nemhauser [1978], Friesz et al. [1989], Ho [1980], Jayakrishnan et al. [1994]). These formulations require attaining the corresponding optimality conditions at every time instant. Most implementations are simulation-based, and the existence of an equilibrium solution has not been analytically proved. However, DTA provides a versatile and coherent framework, capable of capturing the system-level impacts of a multitude of traffic management strategies. The availability of several DTA software packages has motivated its increased utilization by researchers and practitioners.

Yoshii [1996] uses simulation to study the impacts of **real time traffic guidance** based on instantaneous and predicted information, finding positive effects in terms of congestion reduction for a range of information accuracy, which increased along with the percentage of guided vehicles. Yang and Koutsopoulos [1996] implement microsimulation to assess of the impacts of route guidance, arriving at similar conclusions.

Mahmassani and Jayakrishnan [1991] utilize DTA to analyze the effects of real-time in vehicle information provision, finding that the largest benefits for the system occur if vehicles changed their a priori routes only in the face of gains larger than 20%.

Yang et al. [2003] analyze optimal **information** provision strategies using a bi-level mathematical program, which in the first level models the optimization criteria corresponding to different players, such as private information provides, network managers, and system users. In the second level a Dynamic Traffic Assignment simulation is conducted to measure the impacts of the strategies specified in the upper level. The model is implemented to the analysis of the impacts of optimal market penetration levels, information acquisition costs, and information reliability on a small network. The results suggest that, under congested situations, the provision of information is desirable has beneficial system impacts.

L.Engelson [2000] studies the impact of **coordinating information** from different sources on the network performance. The model presented in this work simulates driver's reaction to different sources of information, including in-vehicle guidance systems, radio messages, and variable message signs. In the context of their study, information coordination occurs when the route recommendations based on a particular source of information is made taking into account driver's reaction to other information source/s. Assuming a user-equilibrium type of behavior, they analyzed the total system travel time for different coordination levels among information sources, finding beneficial but relatively low impacts on small, uncongested networks. However, the measured impacts became more noticeable for increasing travel demand and market penetration levels of the route-guidance system.

DTA models provide a very powerful tool to analyze a number of traffic management strategies, as well as the expected behavior of the network under a variety of conditions. However, the corresponding formulations are less transparent, which makes them less appealing as a tool to analyze theoretical advantages and properties of novel approaches to the utilization of information.

3.3 Optimizing information collection and distribution

The collection and distribution of information conditions its availability, and thus its utilization and the corresponding impacts on the system performance. Most of the research efforts in the field of information collection are focused on improving the capability of system managers and operators to measure system parameters and monitor its performance. Active areas of research include the prediction of travel time based on sensor speed measurements or vehicle counts (e.g Ruiz Juri et al. [2007], Foo et al. [2006]), and the estimation of origin-destination demand based on sensor counts (e.g. Bianco et al. [2001], A.Ehlert et al. [2006], Fei et al. [2007]). These works are very valuable, and provide the basis to generate good knowledge base for the planning, management and operation of networks.

The collection of information from **static sensors**, as well as the utilization of wireless location technologies to generate traffic data from p**robe vehicles** of different types are analyzed in Sections 4.1 and 6.1, respectively. Existing methodologies to collect information utilizing fixed and mobile sensors have been typically focused on monitoring the system state, and rarely optimized based on their impact on system performance. The approach proposed int this dissertation for the deployment of sensors explicitly considers the utilization of information for routing purposes, which has the potential to improve the performance of assignment problems in a number of specific applications. Furthermore, the information-based system optimum approach presented in this work provides insights into the design of probe deployment strategies, which has not been studied in the context of transportation problems.

The distribution of information may be accomplished in multiple ways, ranging from general radio broadcasts to the provision of specific route recommendations by in-vehicle route guidance systems or Dynamic Message Signs, discussed in the previous section. Few of the existing models analyze the impact of the location of the DMS on the system performance under information, which would be of great interest.

The reminder of this section discusses an alternative approach to generating and distributing information, given by the zero-infrastructure paradigm, which aims to generate and propagate traffic data by taking advantage of **vehicle-to-vehicle** communication capabilities. Research efforts in this emerging field depart from the assumption that vehicles can communicate information they collect on their paths to other vehicles when their paths cross. Vehicles may exchange information with peers traveling in the same direction or on the opposite lane, and they may re-broadcast the information they receive.

The later is denoted relay communication, and although slower than direct communication, is effective even at low traffic densities (Ziliaskopoulos and Zhang [2003]). A number of practical implementation issues, involving hardware and software design, and information transmission protocols, pose considerable challenges to the implementation of the vehicle -to-vehicle vision. Crucial in an effective implementation of this paradigm are the level of market penetration of equipped vehicles and the range of the wireless communication capabilities, which combined with the prevalent traffic density define how fast and far information can travel. Other important problem variables parameters are the maximum number of vehicles which may communicate simultaneously, the frequency of the information broadcasts, and the time period during which the information is stored and broadcasted. Shladover et al. [2007] conduct a simulation study to analyze the impact of market penetration, traffic density and wireless range on the speed at which messages are propagated. They conclude that low market penetration levels may lead to very slow message propagation speeds, which may discourage the usage of the system at its initial stages. Yang and Recker [2005] use microsimulation to study the propagation of incident data via vehicle-to-vehicle communications, and reach similar conclusions. They suggest the integration of vehicle to vehicle systems with vehicle to infrastructure communication in order to attain larger benefits. They also remark the importance of developing methodologies which vehicles can utilize to estimate the system state based on distributed data. The later is of the utmost importance, and may be challenging, given that the system performance under vehicle-to-vehicle information propagation may differ considerably from the traditional equilibrium assumptions (Ziliaskopoulos and Zhang [2003]) . Jin and Recker [2006] propose an analytical stochastic model to study the probability that a message is propagated further than a given threshold. Their analysis of various scenarios, including incidents, suggests that 7 kilometers is the maximum distance a message may travel. The later is consistent with the results from simulation studies conducted by Yang [2003]. Jerbi et al. [2007] analyze the estimation of traffic densities based on the data propagated by vehicles, and introduce a methodology which leads to fast and accurate estimations of the desired parameters. Wang [2007] proposes closed formulations for estimating the expected value and variance of the propagation distance using the relay methodology, assuming that equipped vehicles arrive into a freeway segment according to a Poisson process.

The vehicle-to-vehicles information propagation approach is of the utmost interest, and while the technology is still under development, it is crucial to develop routing procedures able to take advantage of the information captured in this novel way.

3.4 Summary

The availability of new sources of real-time information invites to think about traditional transportation and network problems from new perspectives. In order to develop and evaluate innovative methodologies, it is crucial to understand the impacts of information at the individual and system level. The review conducted in this section summarizes existing approaches to modeling the utilization of information by drivers and the corresponding effects on the transportation network.

At the individual level, the literature provides models capable of reflecting the use of self-collected and system-level information. The fist type of methodologies captures the adaptive behavior of drivers in the face of the different cost realizations they may observe in a stochastic network. The proposed modeling frameworks and solution techniques, including dynamic programming and heuristic approaches, constitute flexible frameworks within which new information provision strategies may be incorporated and analyzed.

The development of models of the optimal individual response to system-level information was mainly motivated by the advent of route-guidance systems. Even though these models provide routes at the individual level, they reflect a decision made based on the system state, and present an opportunity to incorporate system-optimality considerations in the route choice process. Among the existing approaches, the control-theory-based implementations of iterative models are a promising approach to produce theoretically sound and deployable solutions.

The system-level impacts of information are basically a consequence of the changes that the data availability introduce into the routing decisions of drivers. The most accurate models in the literature build on the methodologies used to describe the individual level behavior, and can be classified in to deterministic and stochastic. Dynamic Traffic Assignment (DTA) models incorporate the time dimension into the concept of user equilibrium (or system optimum), implementing time-dependant shortest path algorithms in the route choice process. These models are a very powerful tool to analyze a number of traffic management strategies, as well as the expected behavior of the network under a variety of conditions. However, the corresponding formulations are not transparent, particularly given the use of simulation. This makes them less appealing as a tool to analyze theoretical advantages and properties of novel approaches to the utilization of information.

Stochastic equilibrium models are based on the concept of adaptive behavior (recourse), and lead to formulations in which the existence and uniqueness of an equilibrium solution may be proved. Even though these methods have not been deployed in large networks, they provide an appealing theoretically sound approach to the analyzed problem, which may be used to better understand the behavior of networks under information. The paradigm of information-based system optimum assignment presented in this dissertation is inspired by models of equilibrium with recourse. However, the explicit consideration of the information collected by assets as they travel through the network leads to fundamentally new formulations.

Finally, the review of existing works in the area of information collection and distribution suggests that major contributions are still possible in that field, particularly if new paradigms for the utilization of information are designed. The models discussed above provide an ideal framework to study optimal information collection and distribution strategies for routing purposes, which has not been accomplished before in the literature. The optimal sensor deployment models presented in this dissertation provide an initial approach to such problem.

Chapter 4

Deployment of Static Sensors for the Support of Adaptive System-Optimum Routing Strategies: Framework

Information is one of the most powerful tools available to mitigate the negative impacts of uncertainty on transportation networks and other stochastic systems. It may be used to enhance the utilization of existing infrastructure, alleviate congestion, and improve safety. Furthermore, the combined consideration of uncertainty and information in decision-support models is critical to generate efficient and robust solutions.

From a decision-making perspective, the value of information highly depends on what information is available (based on spatial and temporal considerations), and how it is utilized. This chapter considers the implementation of information to the support of adaptive system-optimum routing decisions, and focuses on identifying optimal data collection strategies towards this end.

In the context of stochastic networks, adaptive system optimum assignment decisions are such that they may be adjusted based on observed network states. Chapter 3 discussed how the concept of adaptive routing has been implemented in the literature to develop route-guidance strategies based on system-level information. Most of the analyzed efforts are centered on the utilization of information to improve the system behavior, but disregarded the effect of the spatio-temporal characteristics of the data on the quality of the solutions. Conversely, models which focus on the collection of data, reviewed in Section 4.1, rarely consider the utilization of information for routing purposes.

This chapter proposes a model to design data collection paradigm which optimizes the performance of adaptive SO assignment strategies. The approach identifies the links of a network with stochastic arc costs on which sensors should be placed in order to minimize the system expected cost under information. The direct impact of monitoring a link is modeled as the resolution of the corresponding cost uncertainty, in virtue of which a set of perceived network states may be generated. For each of these states it is possible to find the optimal SO assignment solution, consisting of the set of paths on which the system's assets are routed. The solution to the optimal sensor deployment problem specifies sensors location along with an hyperpath describing the optimal SO solution under every perceived state. The proposed models have multiple potential applications, including the routing of special assets, such as emergency response vehicles, or the design of route guidance strategies under extreme circumstances (e.g. an evacuation procedure) in which drivers may be compelled to take routes which do not necessarily maximize their own benefit. Furthermore, the models introduced here may be applied to the optimization of networks representing other systems susceptible to a cooperative behavior.

The optimal sensor deployment model is formulated in Section 4.2. Section 4.3 discusses the expressions for the marginal value of information derived from the proposed formulation, and Section 4.4 analyzes other interesting model properties. These suggest that the novel models are able to capture the non linear impacts of information on the system performance, and contribute to an improved understanding of the problem characteristics. In view of the combinatorial nature of the proposed formulations, their solution poses considerable challenges. Section 4.5 discusses several possible solution approaches and briefly introduces the methodology adopted for this application, which is presented in detail in Chapter 5.

4.1 Optimal sensor deployment on stochastic networks: a literature review

For transportation applications, the study of optimal sensor placement strategies has been traditionally focused on improving the system-monitoring capability for purposes such as the estimation of origin-destination trip matrices (Teodorovic et al. [2002], Ran et al. [2006]) or the collection of Intelligent Transportation Systems (ITS) data (e.g. Sherali et al. [2006]). This section discusses some of the existing approaches to the optimization of static traffic sensor location, while Section 6.1 is concerned with the dynamic collection of such data utilizing appropriately equipped vehicles as probes.

Most of the works reviewed in this section aim to optimize the deployment of sensors in order to better monitor/predict a system parameter, such as the OD trip table, or a performance-related measure, such as system travel time. Modeling the impacts of information on the quality of the measures or predictions may be complex, and model formulations typically adopt a simplified approach, approximating the desired parameter by measures of spatial coverage (Teodorovic et al. [2002]), captured traffic volumes (Sherali et al. [2006], Bianco et al. [2001]), variability of the measured links, or a combination of the former. Very few works take a step further to analyze the impacts of the improved monitoring/prediction capabilities on the system performance.

Thomas [1999] studies the impact of different sensor location strategies in the accuracy of **travel time predictions** on arterial streets using CORSIM, an established micro simulator. Properties such as link travel time and speed are inferred using simple regression models based on a single detector reading. The approach simulates traffic and compares the model fit for different positions of a sensor within a link. This work extends the research by Sisiopiku et al. [1994], focused on finding correlations between various detector readings and link performance measures.

Ruiz Juri et al. [2007] propose a statistical/simulation-based approach to evaluate the effect of sensor location on travel time prediction accuracy. This work explicitly models the impact of specific sensor configurations on the accuracy of the travel time predictions obtained through a methodology that uses cell-transmission based simulation to propagate the traffic counts measured (and predicted) at freeway entry points.

Sherali et al. [2006] analyze the location of Automatic Vehicle Identification (AVI) readers in order to improve travel time predictions. Similarly to Yang and Miller-Hooks [2002], they consider that the benefits derived from placing a sensor on a particular link are a function of the demand and travel time variability affecting all the OD pairs which utilize that link. They assume that each OD pair is connected by a single route, and assign link-dependent AVI reader installation costs. They formulate the problem as a discrete quadratic program, which maximizes the benefits of information, constrained by the maximum number of available readers and a monetary budget. Their exact solution methodology is based on a reformulation-linearization technique, previously introduced in Sherali and Adams [1990], and the incorporation of semi definite cuts, as described in Sherali and Fraticelli [2002].

Bianco et al. [2001] propose a heuristic model to study optimal traffic sensor placement for link flow estimation. The approach minimizes the number of sensors necessary to identify the flows on every link, assuming that the turning percentages at network nodes are exogenously provided. The ultimate goal of this work is the improvement of **OD matrix estimations**, and the authors prove that their methodology leads to bounded estimation errors.

For a similar purpose, Ran et al. [2006] introduce a bi-level model which first deploys sensors on the arcs more likely to capture changes in the demand pattern. The remaining detectors are deployed seeking to maximize spatial coverage. The approach implicitly assumes that measuring those links which flows are more responsive to demand changes leads to more accurate OD matrices estimation.

Teodorovic et al. [2002] study the optimal location of AVI detectors for OD matrix estimation. The genetic algorithm approach proposed in this effort maximizes a function combining OD coverage and the total number of AVI readings.

Yang and Miller-Hooks [2002] introduce a model to locate sensors in a stochastic time-varying network in such way that the benefits of information are maximized. The benefits of information are measured in terms of the number of users for which the travel time uncertainty is reduced, which is approximated by the product of traffic flow and travel time variance on every link. The problem is formulated as a dependent maximum set covering problem, which the authors prove to be NP hard. The dependency is a consequence of explicitly consideration the indirect benefits experienced by drivers using links adjacent to those which are measured. The methodology is implemented using a heuristic approach, and used to find optimal sensor locations on a stochastic, time-varying version of Texas highway network. The authors test the system performance under the information provided by optimally located sensors using an adaptive routing algorithm, described in Miller-Hooks and Mahmassani [2000], Miller-Hooks [2001]. The adaptive routing strategies generated by the algorithm were optimal in most cases, even when a fairly low percentage (30%) of the most-likely used links was instrumented.

The presented review suggests that most of the existing research efforts dealing with optimal sensor placement focus on improving system-monitoring capability. Yang and Miller-Hooks [2002] propose one of the few approaches considering the impact of sensors location on the system performance. However, their methodology optimizes an approximate measure of the benefits of information, rather than modeling the impacts of information on routing behavior. The approach introduced in this chapter explicitly models adaptive routing behavior based on information corresponding to specific network links, thus capturing the complex relationship between sensors location and system performance. The novel model has the potential to improve our understanding of the nature of information impacts, leading to more efficient information distribution and utilization schemes.

4.2 Problem formulation

The problem discussed in this section finds the deployment of K sensors on a network with stochastic arc costs such that the cost of performing an adaptive system-optimum assignment of v assets is minimized. Sensors capture the cost realization on the links they measure, generating a set of **"perceived" network state**s. The System Optimum (SO) assignment solutions are adjusted for each of these states, thus reducing the total system expected cost faced by the system assets with respect to a no-information scenario. The problem solution consists of the set of links on which sensors are deployed, and a hyperpath describing the SO solution under each perceived network state.

Link costs are defined by discrete probability distributions, and they are assumed to be independent of the corresponding flows. If a link is not monitored by a sensor, it is assigned a deterministic cost equal to the expected cost of the corresponding probability distribution. The effect of placing a sensor on a specific link is modeled as a change in the corresponding cost under possible state. Each possible combination of observed states across monitored links generates a "perceived" network state, under which the cost on some links remain uncertain.

The problem lends itself to be formulated as a two level stochastic program, which first level represents the sensor deployment decision, performed under uncertainty. The second level models the optimal routing of the v assets, given the perceived network state measured by the sensors, which information partially resolves the system uncertainty. Given the absence of restrictions on link capacities, and in virtue of the assumed linear cost structure, the optimal route for all system assets under every perceived state is equivalent to the corresponding shortest path. The later, in combination with the fact that the sensor deployment cost is neglected, allows to solve the second level program assuming $v = 1$.

Consider a network $G(N, A)$, where N and A represent the sets of nodes and arcs, respectively. Define $|A| = m$, $|N| = n$, and let ij $s.t.$ $i,\ j \in N$, $i \neq j$ be the links in A, characterized by an infinite capacity and random weights $\tilde{c}_{ij}$. Assume that the latter are independent of the corresponding link flows, and that they follow a discrete probability distribution consisting of a finite number of states $s_{ij} \in S_{ij}$, with probability of occurrence $p_s :\ \sum_{s \in S_{ij}} p_s = 1\ \forall\ ij \in A$. For notational simplicity, the subscript in s_{ij}will be suppressed whenever it can be inferred from the context. Let $r \geq |S_{kl}|\ \forall kl \in A$ represent the maximum number of states observed across all links. Define $c^s_{ij}$the cost realization corresponding to state $s \in S_{ij}$, and denote $\mu_{ij} = \sum_s p_s \cdot c^s_{ij}$ the expected cost of a link $ij \in A$. Network states are a result of the corresponding link states, and are represented using $m-$ dimensional vectors, $\mathbf{w} \in \mathbf{W}$. Let $s^{\mathbf{w}}_{ij}$ be the state on link ij corresponding to network state $\mathbf{w}$, and $c^{\mathbf{w}}_{ij} = c^{s^{\mathbf{w}}_{ij}}_{ij}$the corresponding link cost. Under the assumption of independent and uncorrelated link cost functions, the probability of a network state can be computed as $p_{\mathbf{w}} = \prod_{ij \in A} p_{s^{\mathbf{w}}_{ij}}$. Notice that under the previous assumption, the cardinality of $\mathbf{W}$ is$|\mathbf{W}| = \prod_{ij \in A} |S_{ij}|$, and it grows exponentially with m.

Equations 4.4 to 4.7 present **Formulation #1**, which is a bi-level stochastic program. First and second stage decision variables are binary, and they represent, the placement of sensors on a link (x_{ij}) and the use of a link by an asset (y_{ij}), respectively.

$$\min_{\mathbf{x}} E[f(\mathbf{x}, \tilde{\mathbf{c}})] \tag{4.1}$$

$$\sum_{ij \in A} x_{ij} = K \tag{4.2}$$

$$x_{ij} \in \{0, 1\} \tag{4.3}$$

$$f(\mathbf{x}, \tilde{\mathbf{c}}) = \sum_{ij \in A} y_{ij} \cdot (\tilde{c}_{ij} \cdot x_{ij} + \mu_{ij} \cdot (1 - x_{ij})) \tag{4.4}$$

$$\sum_{ij \in A} y_{ij} + \sum_{ji \in A} y_{ji} = b_j \,\forall\, j \in N \tag{4.5}$$

$$y_{kl} \in \{0, 1\} \tag{4.6}$$

$$b_l = \begin{cases} -1 & \text{if } j \text{ is the destination} \\ 1 & \text{if } j \text{ is the destination} \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

The second level objective function (4.4) represents the summation of the costs paid by the vehicles along the path defined by y_{ij}, which is equal to the cost realization in those links equipped with sensors, and to the link expected cost for the remaining links. Equation 4.5 is the flow conservation constraint, which forces the y_{ij}variables to lay on a path connecting an origin and destination. The number of sensors to be placed is fixed by equation 4.2, which will be referred to as the cardinality constraint. Notice that if $K = 0$, $x_{ij} = 0 \forall ij \in A$, and the problem reduces to a shortest path computation on a network with links cost equal to μ_{ij}. This reflects the fact that, in the absence of further information, the optimal routing strategy is to assign all vehicles to the least-expected cost path. We denote such path $\mathcal{L}^0$, and its corresponding cost ρ^0.

The former formulation can be collapsed into a single level program , with an objective given by equation 4.8, and subject to cardinality constraints (4.2), integrality constraints (4.3, 4.6) and flow conservation constraints

$\sum_{i\in N} y_{ij}^{\mathbf{w}} + \sum_{i\in N} y_{ji}^{\mathbf{w}} = b_j \, \forall\, j,\, \mathbf{w}$. An additional super index is introduced for flow variables, to distinguish them across network states.

$$min \sum_{\mathbf{w}\in\mathbf{W}} p_{\mathbf{w}} \cdot \sum_{ij\in A} y_{ij}^{\mathbf{w}} \cdot (c_{ij}^{\mathbf{w}} \cdot x_{ij} + \mu_{ij} \cdot (1 - x_{ij})) \tag{4.8}$$

Notice that the objective function is non linear and integer, and that the number of flow conservation constraints is $n \times r^m$, which grows exponentially with m. The one-level program may be regarded as a pseudo-boolean problem. In the general case, such problems belong to the NP-hard set, and therefore are not likely to be polynomially solvable Boros and Hammer [2002]. For a rigorous definition of the NP-hard set the reader may refer to Ahuja et al. [1993] and Korte and Vygen [2000].

Equations 4.9 to 4.14 introduce **Formulation #2**, which is based on the definition of perceived network states, and presented for the special case of $z = 1$. It exploits the fact that, form a decision making perspective, only the states at the links equipped with sensors are relevant, which reduces the number of states and variables to be considered. This formulation provides the basis for the solution methodology presented later.

$$\min_{\mathbf{x}} \sum_{ij\in A} x_{ij} \cdot E_{\tilde{c}_{ij}}\left[f_{ij}(\tilde{\mathbf{c}})\right] \tag{4.9}$$

$$\sum_{ij\in A} x_{ij} = 1 \tag{4.10}$$

$$x_{ij} \in \{0,1\} \tag{4.11}$$

$$f_{ij}(\tilde{\mathbf{c}}) = \min_{\mathbf{y}^{ij}}(y_{ij}^{ij} \cdot c_{ij} + \sum_{kl\in A, kl\neq ij} y_{kl}^{ij} \cdot \mu_{kl}) \tag{4.12}$$

$$\sum_{kl\in A} y_{kl}^{ij} + \sum_{lk\in A} y_{lk}^{ij} = b_l \,\forall\, l \in N,\, ij \in A \tag{4.13}$$

$$y_{kl}^{ij} \in \{0,1\} \tag{4.14}$$

The second level objective function in this context is computed for each possible state of every link $ij \in A$, demanding for an increased number of second level variables. The new super-index indicates the sensor providing the information based on which the variable is evaluated.

Similarly to the previous case, the formulation can be collapsed into the single level program presented in equations 4.15 to 4.18.

$$min \sum_{ij \in A} \sum_{s \in S_{ij}} p_s \cdot y_{ij}^{ij,s} \cdot c_{ij}^s + \sum_{kl \in A, kl \neq ik} y_{kl}^{ij} \cdot \mu_{kl} \tag{4.15}$$

$$\sum_{ij \in A} x_{ij} = 1 \tag{4.16}$$

$$\sum_{kl \in A} y_{kl}^{ij,s} - \sum_{lk \in A} y_{lk}^{ij,s} = b_l \cdot x_{ij} \; \forall \, l \in N, \, ij \in A, \, s \in S_{ij} \tag{4.17}$$

$$x_{ij}, \; y_{ij}^{kl,s} \in \{0,1\} \; \forall \, l \in N, \, ij \in A, \, s \in S_{ij} \tag{4.18}$$

For the deployment of a single sensor, the total number of variables and equations in this problem is $n \times m \times r$, smaller than $n \times r^m$ for all values of $1 \leq r < m$. However, the number of variables grows rapidly for an arbitrary value of z. In effect, the consideration of multiple sensors demands to enumerate all the possible ways in which their locations can be selected, as given by the number of combinations of z elements out of m, $C_m^z = \frac{m!}{z!(m-z)!}$.

For the case of multiple sensors, the formulation retains the same structure. Denote $\theta \in \Theta$ each possible combination of z elements out of m, and let δ_θ^{ij} be the link-combination incidence coefficient, equal to one if link ij is in combination θ, and to zero otherwise. Equation 4.19 presents the objective function for such formulation, which is subject to the same constraints as the previous one if the first level decision variables x_{ij} are replaced with x_θ. Each state $s \in S_\theta$ is a combination of the state observed at links $ij \in \theta$.

$$min \sum_{\theta \in \theta} \sum_{s \in S_\theta} \sum_{ij \in A} y_{ij}^{ij,s} \cdot (c_{ij}^s \cdot \delta_\theta^{ij} + \mu_{kl} \cdot (1 - \delta_\theta^{ij})) \tag{4.19}$$

The formulations presented in this section correspond to very large integer programs. Section 4.5.1 discusses possible approaches for their efficient solution. These models are useful to understand the problem structure and properties, which are analyzed in the following sections.

4.3 The marginal value of information

Using finite differences on equation 4.8, one may analyze the marginal impact of information on the system performance according to equation 4.20.

$$\frac{\Delta f(x, y^{\mathbf{w}}(x))}{\Delta x_{kl}} = \frac{\Delta f(x, y^{\mathbf{w}}(x))}{\Delta x_{kl}} \cdot y_{kl}^{\mathbf{w}}(x_{kl}) + \sum_{ij \in A} \frac{\Delta f(x, y^{\mathbf{w}}(x))}{y_{ij}^{\mathbf{w}}(x_{kl})} \cdot x_{kl} \cdot \frac{\Delta y_{ij}^{\mathbf{w}}(x_{kl})}{\Delta x_{kl}} \tag{4.20}$$

The later represents a combination of well known derivation rules (product rule and chain rule), which takes into account that $y_{kl}^{\mathbf{w}}$ (the optimal regular asset assignment for a given information set) is a function of the sensor deployment strategy x_{kl}. The marginal cost, given by Equations 4.21 and 4.22, defines the change in the objective function obtained by placing a sensor on link kl, assuming that the link was previously unmeasured.

$$\begin{aligned} \frac{\Delta f(x, y^{\mathbf{w}}(x))}{\Delta x_{kl}} &= \sum_{\mathbf{w} \in \mathbf{W}} p_{\mathbf{w}} \cdot y_{kl}^{\mathbf{w}} \cdot c_{kl}^{\mathbf{w}} + \sum_{\mathbf{w} \in \mathbf{W}} p_{\mathbf{w}} \cdot \sum_{ij \in A} \frac{\Delta y_{ij}^{\mathbf{w}}}{\Delta x_{kl}} \cdot c_{ij}^{\mathbf{w}} \cdot x_{ij}^{0} \\ &+ \sum_{\mathbf{w} \in \mathbf{W}} p_{\mathbf{w}} \cdot \sum_{ij \in A} \frac{\Delta y_{ij}^{\mathbf{w}}}{\Delta x_{kl}} \cdot \mu_{ij} - \sum_{\mathbf{w} \in \mathbf{W}} p_{\mathbf{w}} \cdot \frac{\Delta y_{ij}^{\mathbf{w}}}{\Delta x_{kl}} \cdot \mu_{ij} \\ &- \sum_{\mathbf{w} \in \mathbf{W}} p_{\mathbf{w}} \cdot \frac{\Delta y_{ij}^{\mathbf{w}}}{\Delta x_{kl}} \cdot \mu_{ij} \cdot x_{ij}^{0} \left(y_{kl}^{\mathbf{w}} \cdot (c_{kl}^{\mathbf{w}} - \mu_{kl}) \right) \qquad (4.21) \\ \frac{\Delta f(x, y^{\mathbf{w}}(x))}{\Delta x_{kl}} &= \sum_{\mathbf{w} \in \mathbf{W}} p_{\mathbf{w}} \cdot \left(y_{kl}^{\mathbf{w}} \cdot (c_{kl}^{\mathbf{w}} - \mu_{kl}) + \frac{\Delta y_{kl}^{\mathbf{w}}}{\Delta x_{kl}} \cdot \mu_{ij} \right) \\ &+ \sum_{\mathbf{w} \in \mathbf{W}} p_{\mathbf{w}} \cdot \left(\sum_{ij \neq kl} \frac{\Delta y_{ij}^{\mathbf{w}}}{\Delta x_{kl}} \cdot \left(c_{ij}^{\mathbf{w}} \cdot x_{ij}^{0} + \mu_{ij}(1 - x_{ij}^{0}) \right) \right) \qquad (4.22) \end{aligned}$$

The first two terms in equation 4.22 represents the impact directly related to the utilization of the link under analysis (local-level impacts of information), if it occurs. The second term captures the impacts resulting from changes in the routing strategy given the new available information (network-level impacts of information). $\frac{\Delta y^{\mathbf{w}}_{ij}}{\Delta x_{kl}}$ implicitly represent these changes, and are equal to zero if the new routing strategy does not affect $y^{\mathbf{w}}_{ij}$, to 1 for those links which are incorporated to the optimal path under $\mathbf{w}$ after placing a sensor in x_{ij}, and to -1 if link ij is removed from the optimal solution given the new information set. The variables x^0_{ij} represent the value of x_{ij} before the incorporation of a sensor on kl. We assume that at most one sensor may be placed at each link, and therefore $x^0_{kl} = 0$.

Notice that the local-level impacts of information may fall within three different categories:

- Measuring impact $(c^{\mathbf{w}}_{kl} - \mu_{kl})$: This type of impact is achieved when link kl was part of the optimal routing strategy under state $\mathbf{w}$ before being monitored, and remains in the corresponding shortest path given the new available information ($y^{\mathbf{w}}_{kl}$=1 and $\frac{\Delta y^{\mathbf{w}}_{ij}}{\Delta x_{kl}}$=0). It reflects an actual change in the expected costs given the new information.

- Incorporation impact $c^{\mathbf{w}}_{kl}$: This impact is attained when link kl enters the optimal solution under $\mathbf{w}$ only after it is assigned a sensor ($y^{\mathbf{w}}_{kl}$=1 and $\frac{\Delta y^{\mathbf{w}}_{ij}}{\Delta x_{kl}}$=1). It simply represents the costs paid by the system assets for utilizing the link.

- Removal impact $-\mu_{kl}$: This impact is a measure of the change in the system cost produce by removing link kl from the optimal solution under $\mathbf{w}$ after placing a new sensor on such link ($y^{\mathbf{w}}_{kl}$=0 and $\frac{\Delta y^{\mathbf{w}}_{ij}}{\Delta x_{kl}}$=-1). It is actually a reflection of the change in the routing strategy brought about by the newly available information.

Using the former definitions, and given the fact that $y^{\mathbf{w}}_{ij}$ is optimally chosen, we may prove that equation 4.23is always true, and therefore the marginal impact of information is always negative (or zero), meaning that it leads to a reduction of the system expected cost.

$$\frac{\Delta f(x_{kl}, y_{kl}^{\mathbf{w}}(x_{kl}))}{\Delta x_{kl}} \leq 0 \tag{4.23}$$

Proof: Let $L_{\mathbf{w}}^{-x_{kl}}$ be the shortest path under state $\mathbf{w}$ before placing a sensor on link kl, and consider two cases for such link: $kl \in L_{\mathbf{w}}^{-x_{kl}}$ (Case I) and $kl \notin L_{\mathbf{w}}^{-x_{kl}}$ (Case IT) .

For Case I, notice first that for all states $\mathbf{w} \in \mathbf{W}^{-}$, where $\mathbf{W}^{-}$is the set of states such that $c_{ij}^{\mathbf{w}} < \mu_{ij}$, the optimal routing strategy will not change as a consequence of the new information (the reader may refer to Section 5.1.3.1 for a proof of this fact). As a result $\frac{\Delta y_{ij}^{\mathbf{w}}}{\Delta x_{kl}} = 0 \; \forall ij \in \mathbf{W}$, which implies that the network-level impacts of information are null, while the local-level impacts, $c_{kl}^{\mathbf{w}} - \mu_{kl} < 0$, are negative.

For the remaining states $c_{ij}^{\mathbf{w}} \geq \mu_{ij}$, and the new routing strategy $L_{\mathbf{w}}^{+x_{kl}}$ may be different from the original one. However, in virtue of the optimality conditions of a shortest path (Ahuja et al. [1993]), Equation 4.24 must hold for every $\mathbf{w} \in \mathbf{W}$.

$$\sum_{ij \in L_{\mathbf{w}}^{-x_{kl}}} y_{ij}^{\mathbf{w}} \cdot (c_{ij}^{\mathbf{w}} \cdot x_{ij} + \mu_{ij}(1 - x_{ij})) \geq \sum_{ij \in L_{\mathbf{w}}^{+x_{kl}}} y_{ij}^{\mathbf{w}} \cdot (c_{ij}^{\mathbf{w}} \cdot x_{ij} + \mu_{ij}(1 - x_{ij})) \tag{4.24}$$

The left hand size of Equation 4.24 represents the cost on $L_{\mathbf{w}}^{-x_{kl}}$, which is a feasible path, and therefore must be greater or equal that the cost along $L_{\mathbf{w}}^{+x_{kl}}$. The elements which are common to both paths may be removed from 4.24, in such way that the left hand side represents the cost of the links removed from the optimal solution, and the right hand side contains the cost of the links incorporated to the optimal solution. Notice that in Equation 4.22, $\frac{\Delta y_{ij}^{\mathbf{w}}}{\Delta x_{kl}} = -1$ for the links in the left hand side, $\frac{\Delta y_{ij}^{\mathbf{w}}}{\Delta x_{kl}} = 1$ for the links on the right hand side, and $\frac{\Delta y_{ij}^{\mathbf{w}}}{\Delta x_{kl}} = 0$ for the remaining links. As a consequence, Equation 4.24 is reduced to $p_{\mathbf{w}} \cdot \sum_{ij \in A} \frac{\Delta y_{ij}^{\mathbf{w}}}{\Delta x_{kl}} \cdot (c_{ij}^{\mathbf{w}} \cdot x_{ij} + \mu_{ij}(1 - x_{ij})) \leq 0$ for $\mathbf{w} \in \mathbf{W}^{+}$, and therefore $\frac{\Delta f(x_{kl}, y_{kl}^{\mathbf{w}}(x_{kl}))}{\Delta x_{kl}} \leq 0$.

For Case 2, a similar approach is applicable. In this instance, kl does not belong to $L_{\mathbf{w}}^{-x_{kl}}$, and the optimal routing strategy remains unchanged for states $\mathbf{w} \in \mathbf{W}^{+}$. Under such conditions $y_{kl}^{\mathbf{w}} = 0$, $\frac{\Delta y_{ij}^{\mathbf{w}}}{\Delta x_{kl}} = 0$, and the local and network-level impacts of information are zero. For states $\mathbf{w} \in \mathbf{W}^{-}$, the same reasoning described above leads to the conclusion that $p_{\mathbf{w}} \cdot \sum_{ij \in A} \frac{\Delta y_{ij}^{\mathbf{w}}}{\Delta x_{kl}} \cdot (c_{ij}^{\mathbf{w}} \cdot x_{ij} + \mu_{ij}(1 - x_{ij})) \leq 0$, which implies $\frac{\Delta f(x_{kl}, y_{kl}^{\mathbf{w}}(x_{kl}))}{\Delta x_{kl}} \leq 0$ under Case II and completes the proof.

The marginal costs described in this section suggest that the benefits of information are accrued when it is possible to take advantage of the fact that $c_{ij}^{\mathbf{w}} < \mu$ in a large number of states, in states with a higher probability of occurrence, or in states exhibiting significant gains $c_{ij}^{\mathbf{w}} - \mu_{ij}$. The feasibility of utilizing the measured links under such conditions depends on the network topology and on the realized/expected cost on the remaining links.

Notice that the computation of the marginal value of information implicitly involves calculating properties of shortest paths on random networks, such as the probability of a link belonging to the shortest path. These properties are very hard to compute, which suggests that the exact solution of the models formulated here cannot be obtained efficiently. Alexopoulous (Alexopoulos [1997]) proves that the evaluation of several of the properties implicitly involved in a marginal cost computation is an #P Hard problem, the equivalent of a NP hard problem for counting problems. Appendix C provides some additional information on this topic.

4.4 Problem properties

This section discusses the problem properties, derived from its mathematical formulation or based on the results of numerical experiments. They illustrate interesting behaviors, which reflect the non-linear impacts of information provision on the system performance.

- The expected cost under information is always smaller or equal than the a-priory expected cost, regardless of the actual cost realizations measured by the sensors. This is a direct consequence of Equation 4.23, proved in the previous section, in view of which the value of information is always non positive.

- Let $E[f^*(K)] = f^*_K$ denote the value of an optimal deployment strategy assigning z sensors to the network. Then $f^*_K \geq f_{perf}$, where f_{perf} is the value of the objective function under perfect information. The later defines the case where the decision maker is aware of the state of the entire network a-priory, and is able to make the optimal decision under any network state. Notice that computing such value may require the enumeration of an exponential number of states.

Proof: The availability of perfect information can be visualized as the deployment of sensors in all network links, including the z optimal links under f^*_z. Given that the marginal value of information is non-positive (Equation 4.23), $f_{perf} \leq \ldots \leq f^*_{z+2} \leq f^*_{z+1} \leq f^*_z$.

- Let θ_z and θ_{z+1} be the optimal sets of monitored links when z and $z+1$ assets are available, respectively. The relationship $\theta_z \subset \theta_{z+1}$ does not necessary hold. The following example illustrates this property, which reflects the non-linear nature of the information impacts.

In Example Network I (Figure 4.1), the default shortest expected cost has a value of 10 units. When a single sensor is deployed, the optimal solution has a cost of 9.5 units, obtained by placing the corresponding detector on link $\boldsymbol{a}$. Placing the sensor in links $\boldsymbol{b}$ or $\boldsymbol{c}$ would report no benefit to the system, given that information from such links not reveal a path cost lower than 10 units under any state. However, when two sensors are available, the simultaneous monitoring of links $\boldsymbol{b}$ and $\boldsymbol{c}$ may detect a path cost realization of only 6 units, which has a 25% probability of occurrence. The former is actually the optimal deployment strategy, leading to a system expected cost of 8.75 units. Any strategy measuring either $\boldsymbol{b}$ or $\boldsymbol{c}$ in combination with other link would not be able to make use of its information, which is worthless if the state at the complementary link remains unknown.

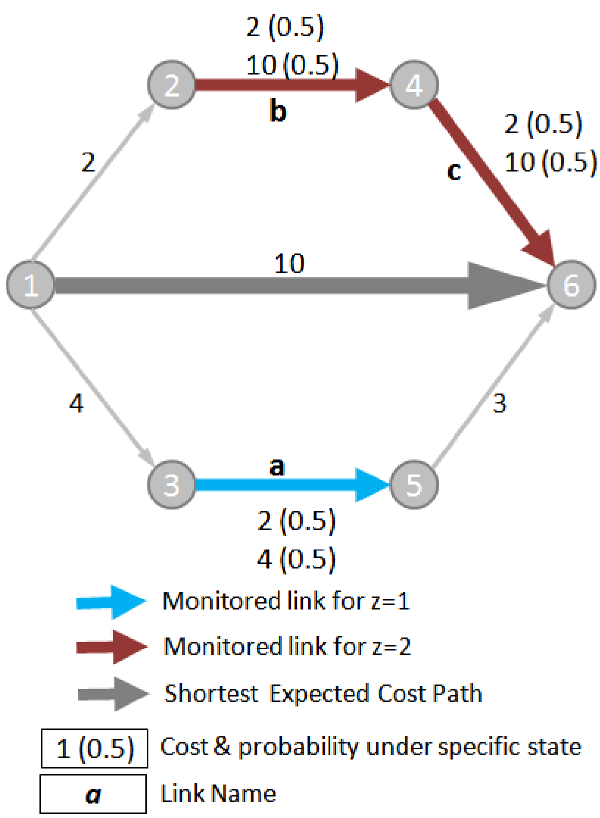

Figure 4.1: Example Network I

- The value of information is not strictly increasing, and it is possible to find cases in which $f^*_{K+1} = f^*_K$, but $\exists\ a > 1 :\ f^*_{K+a} < f^*_K$.

Example Network II (Figure 4.2) presents one of such instances. Similarly to what we observed on the previous example, the information collected from specific sets of links may be complementary, as a result of their relative location and corresponding probability distributions. When this is the case, the benefits of information collection may not be accrued unless all links in the set are measured. In Figure 4.2, a single sensor is optimally located on link $\boldsymbol{a}$, which leads to a system expected cost under information of 11 units, smaller than the default value of 12 units. An additional sensor placed on either of the remaining links would not lead to any further benefits, given that the cost on link $\boldsymbol{d}$ is always lower than the expected cost of $\boldsymbol{b}+\boldsymbol{c}$, provided that only one of them is monitored. As a consequence, the new sensor would not be able to unveil an alternative path cheaper than 16 units under any state. However, if two additional sensors are available, the joint monitoring of links $\boldsymbol{b}$ and $\boldsymbol{c}$ can benefit the system, leading to an expected cost under information of 10.875 units.

- The value $\delta_{max}(ij) = \sum_{\mathbf{w}\in\mathbf{W}^-} p_{\mathbf{w}} \cdot (c^{\mathbf{w}}_{ij} - \mu_{ij})$, where $\mathbf{W}^-$is the set of states $\mathbf{w}$ such that $c^{\mathbf{w}}_{ij} < \mu_{ij}$, defines the maximum benefit which may be accrued by collecting information from link ij. This bound can be attained only when the optimal routing solution under every network state $\mathbf{w} \in \mathbf{W}^-$ utilizes link ij, and provided that the cost of the shortest path connecting i and j when no information is available is greater or equal than μ_{ij}, and when the optimal solution under every state $\mathbf{w} \in \mathbf{W}^-$ uses link ij.

Proof: Consider two cases, Case I, under which $ij \in \mathcal{L}^0$, where $\mathcal{L}^0$ is the shortest path when no information is provided, and Case II, such that $ij \notin \mathcal{L}^0$. Let F_{s-i} and F_{s-j} be the shortest paths connecting the origin to node i, and node j to the destination under state $\mathbf{w}$, respectively. We denote $\mathcal{L}^{\mathbf{w}}$ the shortest path under state $\mathbf{w}$, and $c^{\mathbf{w}}(\mathcal{L}^{\mathbf{w}})$ the corresponding cost. The benefit

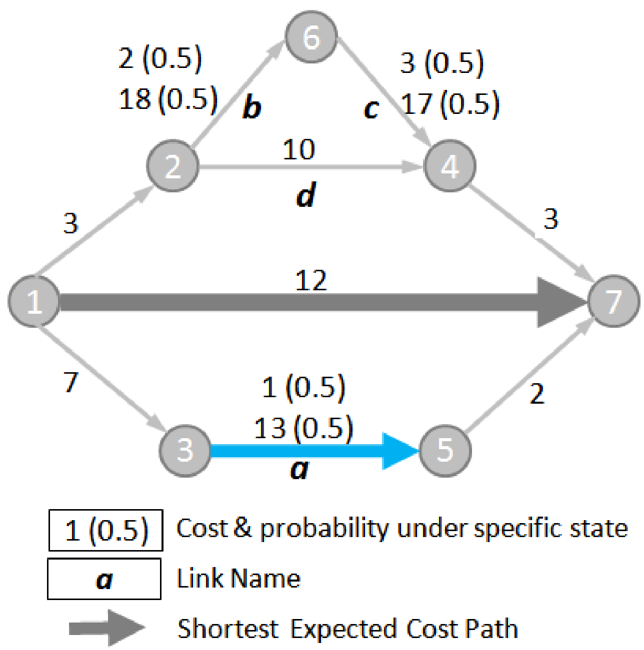

Figure 4.2: Example Network II

of collecting information from link ij is given by $\Delta\rho = \sum_{\mathbf{w}\in\mathbf{W}(ij)} \Delta^{\mathbf{w}}\rho = \sum_{\mathbf{w}\in\mathbf{W}(ij)} c^{\mathbf{w}}(\mathcal{L}^{\mathbf{w}}) \cdot p_{\mathbf{w}} - \rho^0$, where $\mathbf{W}(ij)$ is the set of perceived states revealed by a sensor placed on ij. Given that ρ^0 may be written as $\sum_{\mathbf{w}\in\mathbf{W}(ij)} \rho^0 \cdot p_{\mathbf{w}}$, it suffices to prove that $c^{\mathbf{w}}(\mathcal{L}^{\mathbf{w}}) - \rho^0 \geq c^{\mathbf{w}}_{ij} - \mu_{ij} \; \forall \; \mathbf{w} \in \mathbf{W}^{-}(ij)$. Notice that when $\Delta^{\mathbf{w}}\rho \geq 0$ information does not provide a benefit from the perspective of the system expected cost, and therefore the corresponding state can be excluded form the bound computation (which leads to a looser bound).

For Case II, if ij is part of the optimal solution under $\mathbf{w}$, Equations 4.25 and 4.26 must hold as a consequence of the shortest path definition.

$$c^{\mathbf{w}}(F_{s-i}) + c^{\mathbf{w}}_{ij} + c^{\mathbf{w}}(F_{s-i}) \leq c^{\mathbf{w}}(\mathcal{L}^0) \tag{4.25}$$

$$c^0(F_{s-i}) + \mu_{ij} + c^0(F_{s-i}) \geq \rho^0 \tag{4.26}$$

Additionally, $c^{\mathbf{w}}(\mathcal{L}^0) = \rho^0$, given that the only link on which the cost changes is $ij \notin \mathcal{L}^0$, thus $c^{\mathbf{w}}(F_{s-i}) + c^{\mathbf{w}}_{ij} + c^{\mathbf{w}}(F_{s-i}) \leq \rho^0 \leq c^0(F_{s-i}) + \mu_{ij} + c^0(F_{s-i})$, which implies $\Delta\rho \geq c^{\mathbf{w}}_{ij} - \mu_{ij}$.

If ij is not part of the optimal solution under $\mathbf{w}$, $\mathcal{L}^0 = \mathcal{L}^{\mathbf{w}}$ and $\Delta^{\mathbf{w}}\rho = 0$.

For Case I, if $ij$is part of the optimal solution it is straightforward to see that $\Delta\rho = c^{\mathbf{w}}_{ij} - \mu_{ij}$. If ij is not in the optimal solution under $\mathbf{w}$, $\Delta^{\mathbf{w}}\rho \geq 0$ given that $\mathcal{L}^{\mathbf{w}}$ is some path which cost has not changed (it does not include ij), and therefore $c^{\mathbf{w}}(\mathcal{L}^{\mathbf{w}}) \geq \rho^0$. .

If multiple links are considered simultaneously, similar considerations lead to the conclusion that $kl \in \mathcal{K}$, $\delta(\mathcal{K}) = \sum_{kl\in\mathcal{K}} \delta_{max}(kl)$ is a loose upper bound on the potential benefits of information.

- There exists a value $K^* \leq m$ such that $f^*_K = f^*_{K^*} = f_{perf} \forall \; K > K^*$. This value is a network property for any given origin-destination pair, and provides a bound to the maximum value of information for the corresponding case. In Example Network III (Figure 4.3) $K^* = 1$, and the minimum system expected cost under information is 6.5. Notice that such value can be achieved by placing only one sensor on link

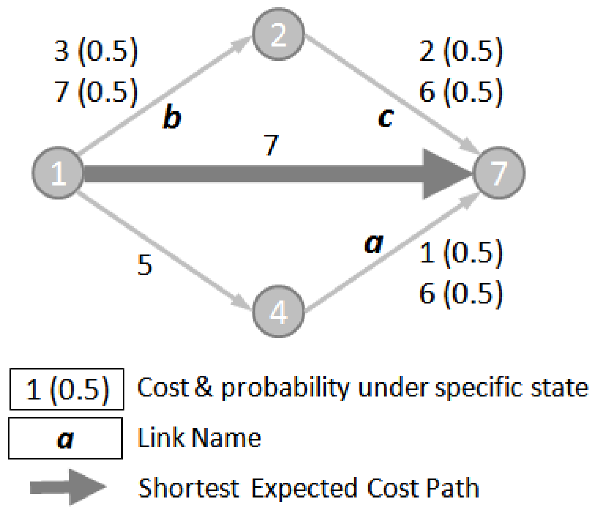

Figure 4.3: Example Network III

a, or by monitoring links ***b*** and ***c*** simultaneously. This illustrates the importance of implementing appropriate models to optimally utilize available resources. Additionally, the identification of the minimum number of links that should be measured in order to attain f_{perf} may pose an interesting topic for further research.

4.5 Possible solution approaches

The formulations introduced in previous sections are integer, and therefore combinatorial in nature. Furthermore, the integrality condition cannot be relaxed for the deployment variable. Mathematically, the later is a consequence of the properties of the constraint matrix, which is not totally unimodular. Intuitively, it is easy to see that if the sensor assignment variable is allowed to take fractional values, then an optimal solution would deploy a fraction of a sensor on every network link, thus achieving benefits similar to those under perfect information.

Mathematical programming techniques, discussed in Section 4.5.1, may be applicable to the problem solution, even though they are likely to be effective for relatively small problem instances.

The methodology developed for the applications analyzed in this dissertation is based on simpler network optimization concepts, ultimately leading to a heuristic implementation. The approach is the flexible to accommodate minor changes in the problem formulation, which may involve a complete re-structuring from a mathematical programming perspective. Additionally, the exact variant is directly linked to the heuristic implementation, which is very convenient from a practical perspective.

The following sections discuss possible mathematical programming approaches to the problem solution, and briefly introduce the methodology adopted for the purpose of this work, which is described and analyzed in Chapter 5.

4.5.1 Exact solution approach

A possible exact solution approach is to consider formulation given by Equation 4.8 as a constrained 0-1 quadratic program. Adams and Sherali (Adams and Sherali [1986]) analyze linearization methodologies to solve similar problems, and Faye and Roupin (A.Faye and Roupin [2007]) introduce a "convexification" procedure which transforms the objective function in order to allow the efficient application of Lagrangian dual approaches.

Another promising way to address the optimal sensor deployment problem solution is to regard it as an uncapacitated network design problem with uncertain arc costs and budget constraints, and solve it by implementing Benders decomposition. Section 4.5.1.2 briefly discusses such technique, while Section 4.5.1.1 presents some basic concepts on stochastic network design.

4.5.1.1 Stochastic network design problems

The deterministic version of the network design problem involves choosing the arcs to be included on a network among a pre-defined set of candidates, in such way that a given origin-destination (OD) flow demand can be satisfied at a minimum total cost. In the general case, such cost is composed by a fixed portion, representing the link installation costs, and a variable portion which depends on the actual flow on each arc. The main problem constraints guarantee flow conservation, in such way that the specified OD demands are satisfied. Additional restrictions to the way in which network arcs may be selected include topological considerations and maximum budget constraints. Costa [2005] provide a fairly comprehensive summary of deterministic network design problem variations, formulations and solution approaches. The network design problem is NP-hard, and exact solution methodologies, which have exponential worse case complexity, include Benders decompositions, Lagrangean relaxation (e.g. Holmberg and Hellstrand [1998]) and dual ascent methods (e.g. Balakrishnan et al. [1989]). In many cases, the exact methodologies are used as the basis for a heuristic approach, in order to solve realistically sized networks.

Uncertainty is incorporated to network design problems searching for more realistic models, at the cost of increasing the complexity of the problem solution. In a stochastic context, the problem objective may be defined from different perspectives, and while some authors propose models which minimize the expected cost of the network design problem, others seek to find "robust" solutions, which attain a minimum performance level under all possible scenarios. Three possible types of uncertainty have been considered, alone or combined, in the existing literature (Costa [2005]): uncertain demands, capacities and arc costs. Not many papers in the literature deal with uncertain arc costs, which is the problem more closely related to the topic of this chapter. Among these, Gutierrez et al. [1996], apply Benders decomposition to a fixed charge network design problem with uncertain arc costs. Their objective is to find a solution such that the routing cost doesn't exceed a threshold value under any scenario. They use a multi-master approach, solving separate master problems for each scenario. Additionally, the authors limit the number of sub problems to be solved at each iteration by carefully analyzing the dual values of each solution, and they add cuts to all the master problems simultaneously.

Following a similar approach, the optimal sensor deployment problem discussed in this chapter may be formulated as an uncapacitated network design problem with uncertain arc costs and budget constraints. In such problem, two types of arcs may be installed between each pair of nodes: one exhibiting fixed costs equal to the expected cost, and one with costs described by a discrete probability distribution. The total number of arcs with stochastic costs is limited by sensors availability, which translates into a budget constraint, and only one type of arc may be installed between two nodes. The solution of such formulation may be approached using Benders decomposition, described in the next section.

4.5.1.2 Benders decomposition

Benders decomposition (Benders [1962]) is a mathematical programming technique used to solve mixed integer problems more efficiently by exploiting special characteristics of their structure. This methodology is appropriate for problems which decompose into a number of simpler sub-problems for fixed values of a subset of the decision variables. When applied to two-stage stochastic programs the methodology is often called "L-shaped method".

Consider a problem with a decomposable structure such as the one presented by 4.27, where x are the continuous decision variables, y are integer decision variables, c and d are the corresponding cost coefficients, and matrices A, B and D represent the problem constraints

$$\begin{array}{cc} \multicolumn{2}{c}{\min_{x,y} cx + dy} \\ Ax + By & \geq b \\ Dy & \geq e \\ x \geq 0 & y \text{ integer} \end{array} \tag{4.27}$$

For fixed values of $y \in Y$, the problem can be expressed by equation 4.28

$$\min_y \{dy + \min_x \{cx : \ Ax \geq b - By\}\} \tag{4.28}$$

The second term can be dualized and, given that it's linear, added to the objective function as displayed in equation 4.29. In this equations u are the dual variables corresponding to the set of constraints $Ax \geq b - By$, and Y is the feasible space for y, as defined by the original constraints.

$$\min_{y \in Y} \; \{dy + \max_{u \geq 0} \{u(b - By) : uA \leq c\}\} \tag{4.29}$$

The solution space of the sub problem, assumed to be nonempty, can be represented in terms of its extreme directions, given by $r^q(b - By) \leq 0 \; \forall q = 1, 2, ...Q$. Similarly, the objective function can be expressed as a function of the corresponding extreme points. The main disadvantage of the later that it leads to a formulation with a very large number of constraints, given by all the extreme directions of the solution space. Nevertheless, such formulation lends itself for an iterative solution approach, in which the extreme directions are added progressively by alternating between the solution of relatively easy sub problems, which provide the dual variables, and a relaxed version of the original problem. Notice that the procedure maintains dual feasibility at all times, and provides the information to generate/update upper and lower bounds for the optimal solution to the original problem which are used to define convergence.

The ability to solve the sub problems efficiently is vital for a successful application of Benders approach. Nevertheless, most of the computational effort is typically devoted to the solution of the master problem (Magnanti and Wong [1981]), particularly as the number of added constraints becomes larger. Some existing approaches tackle this problem by reducing the number of active constraints (typically called cuts) at each iteration (e.g. Marín and Jaramillo [2008]), by carefully selecting the cuts to add in the search for tighter bounds (e.g. Magnanti and Wong [1981]), and even by allowing a sub-optimal solution to the master problem (e.g. Burkard and Bonniger [1983]).

Even though Benders decomposition may, in the worst case, involve solving the full mathematical problem, it has been found to perform very efficiently for particular types of problems, including network design (Geoffrion and Graves [1974], Magnanti et al. [1986]). An additional advantage of this methodology is that the continuous provision of upper and lower bounds allows defining sub-optimal termination criteria when appropriate.

Implementing Benders decomposition to the stochastic network-design version of the problem considered in this chapter would involve the solution of a sub-problem for each perceived network state at every iteration. Perceived states are defined based on the sensor deployment strategy obtained from the solution of the relaxed master problem, and the corresponding sub problem is a shortest path problem, which provides the dual variables necessary to incorporate new constraints (cuts) to the master problem. It is interesting to notice that, form an implementation perspective, the methodology would not be very different form the solution approach presented in Chapter 5.

4.5.2 Implemented heuristic solution approach

The solution methodology, described and implemented in chapter 5, is based on the observation that the problem posed in 4.2 may be solved by complete enumeration, computing the shortest path between origin and destination for each possible network state. This would involve $m \times r$ computations for the deployment of a single sensor, and a combinatorial number of operations in the general case, where the set of possible strategies is given by $C_z^m = \frac{n!}{(m-n)!n!}$, and the number of shortest path computations is in the order of r^z. Shortest path computations can be executed very efficiently, applying various implementations of the well-known Dijkstra's algorithm (Ahuja et al. [1993]). Moreover, if we consider a single OD pair case, even faster algorithms, such as A* (Klunder and Post [2006]) may be implemented. In medium size network the number of computations required within a complete enumeration framework easily reaches the order of billions. This limits the applicability of the exact approach and motivates the methodologies developed in the next chapter, which heuristically reduce the number of strategies to evaluate utilizing a Tabu search approach. The evaluation of each strategy is efficiently achieved by implementing a state-space partitioning technique, in virtue of which it is not necessary to compute the shortest path under every perceived network state. The integrated approach is flexible, and numerical results suggest it is effective for the solution of a range of problems, even though the exact evaluation of feasible strategies may limits its applicability, particularly on poorly connected networks.

4.6 Summary

This chapter formulates and discusses the optimal deployment of static sensors for the support of adaptive System-Optimum (SO) routing strategies. The proposed model selects the location of a fixed number of static sensors, z, leading to the minimum system expected cost under information. This approach differs from existing efforts in the literature, which typically analyze the collection of information from the perspective of improving the capability to monitor system parameters or performance measures. The novel paradigm explicitly considers the impacts of system-level information on routing decisions. Unlike most of the existing research on the field, the proposed approach takes into account the relationship between the location from which information is collected and the resulting system performance.

In the context of this chapter, the local-level impact of information provision is modeled as a change on the cost at the monitored links, which leads to a set of perceived network states. Under an adaptive routing paradigm, SO assignment decisions may be adjusted for each perceived states, leading to an improved expected performance.

Section 4.2 presents three alternative mathematical formulations of the proposed problem. Even though these formulations are combinatorial in nature, they are useful to understand the problem properties, and can be utilized as the basis for efficient solution methodologies. They also provide a means to derive a theoretical expression for the marginal value of information (Section 4.3), which is proved to be always positive under the adopted assumptions. The marginal cost formulation is a valuable tool to understand the models behavior, and to interpret problem properties, presented in Section 4.4. The observed properties illustrate interesting model's behavior, and reflect the non-linear nature of the impacts of information. For example, the incorporation of additional sensors may not improve the system expected cost. Conversely, "synergic" effects can be observed, in virtue of which the benefits obtained by jointly monitoring of subset of links is greater than the improvements accrued by placing a single sensor in any link in the set.

The exact problem solution is discussed in Section 4.5, which summarizes possible mathematical programming approaches, including Benders decomposition and quadratic programming techniques, that may me applicable. However, these approaches are likely to be effective only on relatively small problem instances, which motivates the choice of a solution methodology based on network optimization methods (Chapter 5). The approach lends itself to a heuristic implementation, and can easily incorporate changes in the problem assumptions and formulations.

The problem described in this section is of interest from a variety of perspectives. Section 7.1 discuses some of its potential applications, which range from the deployment of sensors during rescue operations, to data filtering for online routing purposes. Section 7.1 also presents desirable model extensions, including sequential deployment strategies and more complex objective accounting for the robustness of the solution.

Chapter 5

Deployment of Static Sensors for the Support of Adaptive System-Optimum Routing Strategies: Methodology and Implementations

The optimal solution to the problem presented in this chapter entails finding a strategy to deploy s sensors on a stochastic network with m edges such that the expected system cost under information provision is minimized. A distribution/deployment strategy is specified by the set of links chosen to place sensors. As a consequence of the binary, non-convex, nature of the expected system cost function , the identification of an optimal solution may demand the evaluation of the C_m^s possible deployment strategies, a number that grows exponentially with the number of sensors and the network size. Furthermore, each of these evaluations implicitly involves the computation of properties of paths in stochastic networks, as discussed in Section 4.3, which has been proved to be very challenging (Appendix C).

From a naive approach, the evaluation of a strategy may require computing a shortest path for every perceived network state given a particular deployment strategy. Despite the availability of very efficient methods for the calculation of shortest paths (Ahuja et al. [1993]), the computational effort required by problems involving large networks, numerous states and multiple sensors may easily become prohibitive, particularly if a solution is needed within a short time frame.

The methodology presented in this chapter takes a two-folded approach to reduce the number of calculations needed to provide an optimal, or near optimal solution, which separately addresses the two major challenges identified above. The number of shortest path calculations required to evaluate a feasible solution is reduced by developing and implementing a state-space partitioning methodology which incorporates some shortest path re-optimization concepts. In order to limit the total number of strategies to be evaluated a Tabu search heuristic methodology is tailored to the problem under study.

The state partitioning approach, described in Section 5.1, has the additional advantage of providing a flexible framework which can accommodate the solution of more complex problem variants, such as those involving flow-dependent or time-dependent arc costs. Tabu search is a meta heuristic procedure, which does not guarantee he optimality of the solution at convergence. However, the implementation developed for this application, described in Section and tested in Section 5.2, consistently provided solutions extremely close to the optimal value. This chapter provides a detailed description of the two components of the proposed solution methodology, which can be used alone or in combination to efficiently solve the problem described in Chapter 4. Sections 5.1.5 and 5.1.5 present the numerical tests conducted to assess the performance of the solution methodology, which is implemented in Section 5.3 to the analysis of the impacts of sensors location on the performance of adaptive SO routing strategies.

5.1 Evaluating a sensor deployment strategy: a state-space partitioning approach

For the optimization problem described in Chapter 4, the evaluation of any specific sensor placement strategy involves determining the system expected cost given the information provided by the deployed sensors. In the context of this problem, information translates into the identification of different cost realizations at monitored links, which combined generate a set of perceived network states. The adaptive routing paradigm allows adjusting the assignment strategies under each perceived state, leading to an improved expected system performance.

A naive approach to compute the system expected cost in this setting is to calculate the optimal routing strategy for any possible perceived network state. However, the problem characteristics allow reducing the total number optimization problems to be solved by implementing a state-space partitioning approach. The procedure is based on the principles presented by Alexopoulos [1997] for the evaluation of properties related to shortest paths on networks with discrete random arc costs. He studied measures such as the probability of a given path being the shortest, or those of the shortest path not exceeding a threshold value, proving that their computation constitutes an #P-hard problem (for definitions and examples of #P-Hard problems, the reader may refer to Valiant [1979a]).

The problems analyzed in Alexopoulos [1997] ultimately entail identifying which network states "contribute" to the computed measure. The later typically involves comparing the value of a given path under a specific state to a threshold value. The fundamental concept behind this approach is that a single shortest path computation can be used to classify entire ranges of network states, defining whether or not they should be considered in the computation of the analyzed measure. By avoiding the evaluation of a new shortest path problem under each possible state, the technique greatly reduces the computational effort.

The solution methodology presented by Alexopoulous iteratively partitions the network state-space, generating bounds for the studied property which improve at each iteration, leading to the exact solution. The efficacy of the procedure depends on the strategy used to subdivide the state-space after each shortest path evaluation, which determines the total number of runs to be performed. The proposed partitioning scheme allowed the solution of problems involving more than 87×10^9 possible states by evaluating only $\sim 12,000$ shortest paths. Furthermore, the nature of the algorithm is such that no more than 11 partitions had to be stored simultaneously at any point.

The methodology discussed above is not directly applicable to the problem discussed in this chapter, mainly due to the more complex nature of the property we seek to evaluate. Computing the expected cost of an adaptive system optimum assignment strategy requires knowing the exact cost on the shortest path under each perceived network state, as opposed to only determining whether such value exceeds or not a given threshold. Even though this naturally increases the number of necessary evaluations, it is still possible to take advantage of simple shortest path properties to reduce the computational burden. Section 5.2.1 presents a state-partitioning scheme based on the same fundamental principles proposed in Alexopoulos [1997], which can be used to efficiently compute the system expected cost under information. Numerical tests conducted on a variety of networks of different sizes and statistical properties (Section 5.1.5) suggest that the proposed methodology is robust, and that it may find the expected cost of an adaptive SO strategy by evaluating less than 10% of the perceived network states.

5.1.1 Methodological framework

This section introduces a state partitioning framework to find $Z^{\mathcal{K}}$, the expected cost of an adaptive system-optimal network assignment problem under the information provided by a pre-defined set of K sensors placed on links $k \in \mathcal{K}$. Even though in the worst case the problem solution may entail finding the shortest path under each possible network scenario, the methodology presented here performs much more efficiently in practice (Section 5.1.5). The notation introduced to describe the algorithm is slightly different from the nomenclature introduced in the previous chapter, and is defined below.

The problem considers a directed network $G(N, M)$, with nodes $i \in \mathcal{N}$, and edges $j \in \mathcal{M}$. We define $|\mathcal{M}| = M$ and $|\mathcal{N}| = N$. Links may be designated also using the indices of the corresponding origin and destination nodes (e.g. ij is the link connecting nodes i and j). The arc costs are random, following discrete probability distributions ξ^jconsisting of S^j states $s^j \in \mathcal{S}^j$. Each state is defined by its cost $\varepsilon^j_{s^j}$ and the corresponding probability, $p^j_{s^j}$, in view of which $\mu^j = \sum_{s^j \in \mathcal{S}^j} \varepsilon^j_{s^j} \times p^j_{s^j}$ is the link expected cost. For every link we assume $\varepsilon^j_1 < \varepsilon^j_2 < ... < \varepsilon^j_s$. Notice that the cardinality of the sets ξ^jcan vary across links.

Any particular combination of link cost realizations originates a network state $x \in \mathcal{X}$. These can be defined by m-tuples $x = \{s^j(x)\}$, $j \in \mathcal{M}$, specifying the state realized at link j under network state x. The corresponding link cost and probabilities are given by $\varepsilon^k_{s^k(x)}$ and $p^k_{s^k(x)}$, respectively. For notational convenience, the later may be denoted by $c^k(x)$ and $p^k(x)$, or c^k_x and p^k_x. Indices $s^j(x)$ can adopt any value in the range $\left\{1, 2, ..., S^k\right\}$, in virtue of which there are $T = \prod_{j \in M} S^j$ possible network states, with a probability of occurrence given by equation 5.1.

$$r(x) = \prod_{j \in \mathcal{M}} p^j(x) \tag{5.1}$$

A path L is a set of links, and its cost under any state is given by the summation of the cost realizations at the corresponding links $c_x(L) = \sum_{j \in L} c^j(x)$.

A partition of the m-dimensional state space $\mathcal{X}$ based on links $k \in \mathcal{K}$ is a subset $\mathcal{X}_1 \subseteq \mathcal{X}$ of the states in such space, defined by two m-tuples $\alpha(\mathcal{X}_1) = \left\{a^k(\mathcal{X}_1)\right\}$and $\beta(\mathcal{X}_1) = \left\{b^k(\mathcal{X}_1)\right\}$, where a^k and b^k indicate the index corresponding to the first and last link state included in $\mathcal{X}_1$. We will extend the definition of $r(x)$ to represent the probability of a subset as indicated in equation 5.2. If the subset contains a single element, $a^k = b^k = s^k \ \forall \ k \in \mathcal{K}$, and equations and 5.1 and 5.2 are equivalent.

$$r(\mathcal{X}_1) = \prod_{k \in \mathcal{K}} \left(\sum_{s^k = a^k(\mathcal{X}_1)}^{s^k = b^k(\mathcal{X}_1)} p_{s^k}^k \right) \tag{5.2}$$

We denote by $K \leq M$ the number of available sensors, which are deployed according to an exogenously determined strategy t. Binary variable $g^j(t)$ is used to identify the links on which sensors are placed under a particular strategy, by setting it to one if a sensor is located on link j under strategy t, and to zero otherwise. Strategy t can be defined by the set of links $k \in \mathcal{K}(t)$, such that $g^k(t) = 1$. For notational simplicity, $\mathcal{K}$ will be used instead of $\mathcal{K}(t)$ when the value of t is obvious given the context. For all links $w \notin \mathcal{K}$ we assume $c^w(x) = \varphi^w$ under any state x. In view of the former, the network states which can be perceived given the information provided by strategy t differ only on the cost realization at the measured links. The set of perceived network states consists of K-tuples $\widetilde{x} \in \mathcal{P}$, such that $\widetilde{x} = \left\{c^k(x)\right\}, \ k \in \mathcal{K}$, and $r(\widetilde{x}) = \prod_{k \in \mathcal{K}} p^k(\widetilde{x})$. The total number of perceived states is given by $P = \prod_{k \in \mathcal{K}} S^k$.

The cost at the shortest path connecting nodes o and d under a general or perceived scenario is ρ_{od}^x. The shortest path computed under a no-information provision scenario $x^0 = \{\mu^j\} \ \forall \ j \in \mathcal{M}$ is the shortest expected cost path, with cost ρ_{od}^0. The set $\mathcal{L}_{od}^0$ contains all the links included in the shortest path. For the examples and descriptions presented in this section, a single origin-destination pair will be considered, and therefore the sub index od will be omitted.

The system expected cost under information set $\mathcal{K}$ is $Z^{\mathcal{K}}$, and it includes the contributions of all the states $\widetilde{x} \in \mathcal{P}$ (Equation 5.3) to the system expected cost, $z^{\mathcal{K}}(\widetilde{x})$. In the general case, the evaluation of $Z^{\mathcal{K}}$ would involve the computation of $\rho^{\widetilde{x}}$ for all the possible perceived states.

$$Z^{\mathcal{K}} = \sum_{\widetilde{x} \in \mathcal{P}} \rho^{\widetilde{x}} \times r(\widetilde{x}) = \sum_{\widetilde{x} \in \mathcal{P}} z^{\mathcal{K}}(\widetilde{x}) \tag{5.3}$$

The state partitioning approach proposed here aims to identify non-overlapping subsets $\mathcal{P}_v \subseteq \mathcal{P}$ such that their contributions can be calculated based on a single shortest path computation, thereby reducing the computational effort. The value of $Z^{\mathcal{K}}$ is then obtained according to 5.4, where $\mathcal{W} \subseteq \mathcal{P}$ is the subset of states $\widetilde{w}$ which do not belong to any partition $\mathcal{P}_v$, and $\mathcal{V}$ is the set of all existing partitions.

$$Z^{\mathcal{K}} = \sum_{v \in \mathcal{V}} z^{\mathcal{K}}(\mathcal{P}_v) + \sum_{\widetilde{w} \in \mathcal{W}} z^{\mathcal{K}}(\widetilde{w}) \tag{5.4}$$

5.1.2 State-space generation: Visualizing the state space as a trees

The order in which the states within a state space are considered plays a fundamental role in the efficiency of any state space partitioning technique. Such order influences the total number of partitions to be generated, the total number of evaluations required, and the number of partitions which need to be stored simultaneously. For the implementation discussed in this paper, the state ordering is given by the state generation methodology, which is presented in this Section. The methodology was designed based on the findings presented in Alexopoulos [1997], and tailored to fit our goal of minimizing the number of shortest path computations required to evaluate the objective function. .

In order to ease the interpretation of the state generation procedure and the various state-space partitioning rules introduced in later sections, one may regard the state space as a set of trees, and the state space partitioning process as a tree-pruning procedure.

Assume that index k, such that $1 < k < K$, is used to identify the links in $\mathcal{K}$, and consider S^1 trees of depth K. In these trees, each node at level $l = (1, 2, ..., K-1)$ has s^{l+1} children , indexed by $w = (1, 2, ..., s^{l+1})$. Additionally, assume that link costs are such that $\varepsilon_1^j < \varepsilon_2^j < ... < \varepsilon_s^j$. Figure 5.1 presents Example 1, which illustrates the tree representation of the state space generated by 4 links. It is easy to see that the number of leaves in such trees is equal to the cardinality of the set $\mathcal{P}$ (32 in our example), and that the corresponding branches represent all possible states. In Figure 5.1 the colored branch represents state $\widetilde{x} = \{1, 1, 3, 1\}$.

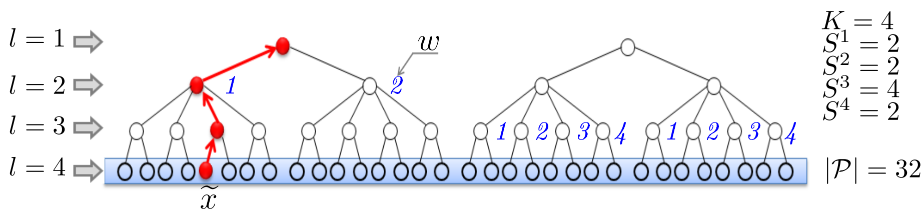

Figure 5.1: Tree representation of a perceived state space

Finding the system expected cost is equivalent to identifying, for every tree leave, the shortest path under the corresponding state along with its probability. The methodology proposed here generates the states in $\mathcal{P}$ according to the pseudo code described by Algorithm 1, which is a modification of the methodology presented by Rosen [1991].

Algorithm 1 State Generation Pseudo Code

---

for $1 \leq l \leq K$ **do**
  $s^k = 1 \; \forall \; k \in \mathcal{K}, \; k \neq l$
  $s^l = 2$
  $\text{max\_index}^k = S^k \; \forall \; k \in \mathcal{K}$
  $\text{min\_index}^k = 1 \; \forall \; k \in \mathcal{K}, \; k \neq l$
  $\text{min\_index}^l = 2$
  end_flag=0
  while (end_flag=0) **do**
    $p = l$
    while $(s^p = S^p$ and $p \geq 0)$ **do**
      $p = p - 1$
    if $(p > 0)$ **then**
      $s^p = s^p + 1$
      for $(p + 1 \leq k \leq K)$ **do**
        $s^k = \text{min\_index}^k$
    else
      end_flag=1

This is equivalent to moving through the trees by "levels" starting at depth 1, and considering the nodes in a level from left to right, as depicted in Figure 5.2. Each level represents a link $k \in \mathcal{K}$, and the values of w at the considered node indicates the link state index corresponding to a particular state. For

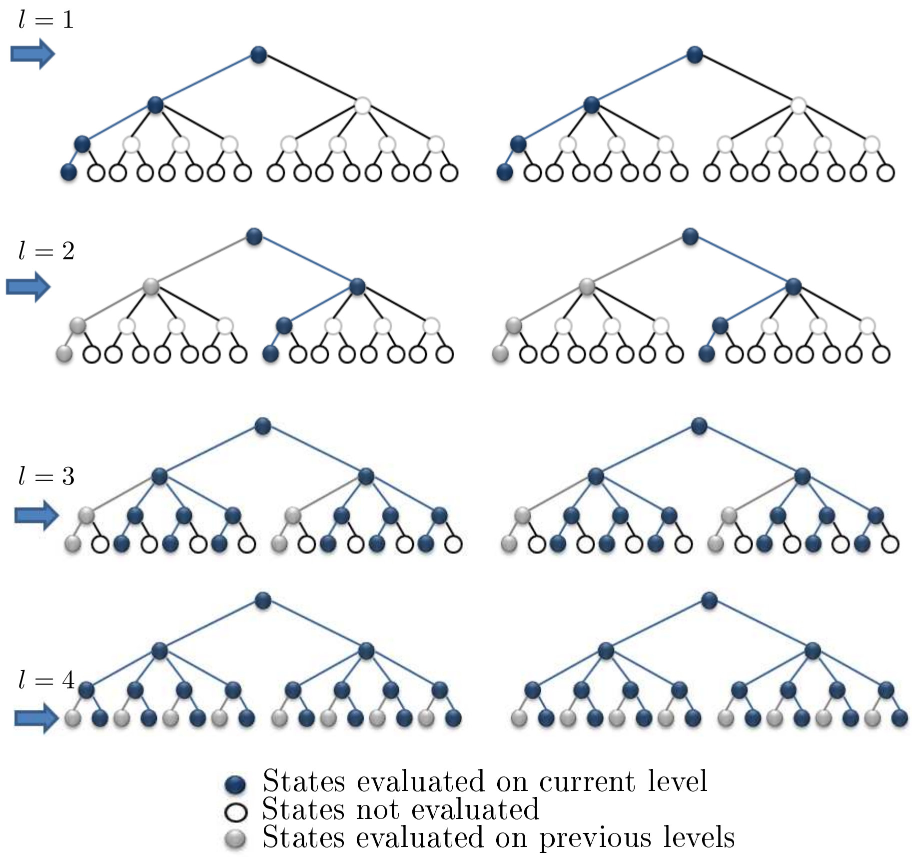

Figure 5.2: Tree interpretation of the state generation procedure

links $k : \; k > l$ the link state index is set to one, in such way that the states generated at level l differ from the ones generated at previous levels only by the cost at link $k = l$.

Table 5.1 presents the order in which the states corresponding to Example 1 are generated, which may be regarded as lexicographic up to the l^{th} index for every level l. Notice that state $\widetilde{x} = \{1, 1, 1, 1\}$ is generated as part of the initialization process, and that at level l the lowest value for s^l is 2, accounting for the fact that states $\widetilde{x} : \; s^l = 1$ have been generated on previous levels. Also notice that algorithm 1 can be used to generate the states within a partition $\mathcal{P}_v$ by setting vectors max_index and min_index equal to $\alpha(\mathcal{P}_v)$ and $\beta(\mathcal{P}_v)$ respectively.

$l=1$	$l=2$	$l=3$	$l=4$
$\{1,1,1,1\}$	$\{1,2,1,1\}$	$\{1,1,2,1\}$	$\{1,1,1,2\}$
$\{2,1,1,1\}$	$\{2,2,1,1\}$	$\{1,1,3,1\}$	$\{1,1,2,2\}$
		$\{1,1,4,1\}$	$\{1,1,3,2\}$
		$\{1,2,2,1\}$	$\{1,1,4,2\}$
		$\{1,2,3,1\}$	$\{1,2,1,2\}$
		$\{1,2,4,1\}$	$\{1,2,2,2\}$
		$\{2,1,2,1\}$	$\{1,2,3,2\}$
		$\{2,1,3,1\}$	$\{1,2,4,2\}$
		$\{2,1,4,1\}$	$\{2,1,1,2\}$
		$\{2,2,2,1\}$	$\{2,1,2,2\}$
		$\{2,2,3,1\}$	$\{2,1,3,2\}$
		$\{2,2,4,1\}$	$\{2,1,4,2\}$
			$\{2,2,1,2\}$
			$\{2,2,2,2\}$
			$\{2,2,3,2\}$
			$\{2,2,4,2\}$

Table 5.1: State space for Example 1

5.1.3 State-space partitioning: Partitioning rules

The naive approach to the system expected cost computation is equivalent to generating and evaluating all the tree leaves (i.e. computing $\rho^{\widetilde{x}}$ for all $\widetilde{x} \in \mathcal{P}$). The goal of the proposed state partitioning methodology is to take advantage of some problem properties in order to evaluate several leaves based on a single shortest path computation, thus reducing the computational effort. This leads to a subdivision of the state space, which distinguishes between subsets of states (represented by tree branches) which have already been evaluated and sets which require further consideration. Each of these sets (or partitions) is analyzed by generating and evaluating the corresponding states, and it may be sub-partitioned.

The circumstances under which more than one leave can be evaluated simultaneously are identified using partition rules. These may be slightly different depending on whether the considered sensor deployment strategy

t involves none (Type I), some (Type II), or all (Type III) the links in the shortest path. The following sections describe the partitioning rules corresponding to each strategy type, and present the algorithmic implementations. Section 5.1.4 presents the final algorithm, and discusses some specific implementation issues.

5.1.3.1 Strategies Type I

For strategies of Type I, on which all sensors are placed outside the shortest expected cost path ($\mathcal{L}^0 \cap \mathcal{K} = \emptyset$), a very efficient state partitioning scheme can be defined, closely related to the one proposed in Alexopoulos [1997]. Given that there no sensors is placed on the shortest expected cost path, the perceived cost at $\mathcal{L}^0$ does not change based on the sensor information. In virtue of this ρ^0 becomes a deterministic upper bound on the optimal cost corresponding to any perceived network realization, which constitutes the basis of the first partitioning rule, based on the following facts:

- **Fact 1:** The shortest path value corresponding to any perceived network state $\widetilde{x} \in \mathcal{X}$ will be lower than ρ^0_{od} only if for at least one $k \in \mathcal{K}$ it is true that $c^k(\widetilde{x}) < \mu^k$. In other words, $\rho^{\widetilde{x}} < \rho^0 \Rightarrow c^k(x) < \mu^k$ for at least one link $k \in \mathcal{K}$. This is a necessary, although not sufficient condition.

Proof: Equation 5.5, where $L_{od} \in \mathbf{L}$ are all paths connecting the origin and destination, reflects the optimality condition defining a shortest path. Given that cost changes are possible only on those links in $\mathcal{K}$, the cost on $\mathcal{L}^0$ is equal to ρ^0_{od} under every state $\widetilde{x}$. If $\rho^{\widetilde{x}} < \rho^0$ the path used to achieve $\rho^{\widetilde{x}}$ is $\mathcal{L}^{\widetilde{x}} \neq \mathcal{L}^0$. Such path was not optimal before the cost change, and therefore its cost must had been greater to, or equal than, ρ^0. In view of the former, $\rho^{\widetilde{x}} < \rho^0$ implies that the cost at some link in $\mathcal{L}^{\widetilde{x}}$ has decreased, thus $\mathcal{L}^{\widetilde{x}}$ contains at least one link $k \in \mathcal{K}$ for which $c^k(\widetilde{x}) \leq \mu^k$ which completest the proof.

$$\rho^0_{od} = \sum_{j \in \mathcal{L}^0} \mu^j \leq c_0(L_{od}) \forall L_{od} \in \mathbf{L}_{od} \tag{5.5}$$

Corollary: The contribution to $Z^{\mathcal{K}}$of states states $\widetilde{y}$ such that $c^k(\widetilde{y}) > \varphi^k \ \forall \ k \in \mathcal{K}$ may be computed according to equations 5.6 and 5.7, without performing a shortest path evaluation. In these equations $\mathcal{Y}^1 = \{\widetilde{y}\}$ denotes the set of all links $\widetilde{y}$.

$$z^{\mathcal{K}}(\mathcal{Y}) = \rho^0 \times r(\mathcal{Y}^1) \tag{5.6}$$

$$r(\mathcal{Y}^1) = \prod_{k \in \mathcal{K}} \left(\sum_{s=\varphi^k}^{s=b(\mathcal{X})} p_s^k \right) \tag{5.7}$$

- **Fact 2:** If $\rho^{\widetilde{x}} = \rho^0$ for a perceived network state $\widetilde{x} \in \mathcal{P}$, then $\rho^{\widetilde{x}} = \rho^0 \ \forall \ \widetilde{y} \in \mathcal{Y}^2 : s^k(\widetilde{y}) \geq s^k(\widetilde{x}) \ \forall \ k \in K$.

Proof: Given that $\rho^{\widetilde{x}} = \rho^0$, $\mathcal{L}^0$ is valid as a shortest path under $\widetilde{x}$, in virtue of which equation 5.5 is valid. Given that state indices are ordered in increasing order of their corresponding costs, the assumption $s^k(\widetilde{y}) \geq s^k(\widetilde{x})$ is equivalent to $c^k(\widetilde{y}) \geq c^k(\widetilde{x})$ which, following the same reasoning described for Fact 1, implies that $c_{\widetilde{y}}(L) \geq c_{\widetilde{x}}(L) \geq c_{\widetilde{x}}(\mathcal{L}^0) \ \forall \ L \in \mathbf{L}$.

Corollary: If the shortest path value under a perceived network state $\widetilde{x} \in \mathcal{X}$ is $\rho^{\widetilde{x}} = \rho^0$, equations 5.8 and 5.9 can be used to compute the contribution to $Z^{\mathcal{K}}$ of all states $\widetilde{y}$ such that $s^k(\widetilde{y}) \geq s^k(\widetilde{x}) \ \forall k \in \mathcal{K}$.

$$z^{\mathcal{K}}(\mathcal{Y}^2) = \rho^0 \times r(\mathcal{Y}^2) \tag{5.8}$$

$$r(\mathcal{Y}^2) = \prod_{k \in \mathcal{K}} \left(\sum_{s=s^k(\widetilde{x})}^{s=b(\mathcal{X})} p_s^k \right) \tag{5.9}$$

In virtue of these facts, we can define the following partitioning rule for deployment strategies of Type 1:

Partitioning Rule 1

Let$\mathcal{X}$ be a K-dimensional space describing the perceived network state-space based on the information provided by K

sensors, or a subset of such space. Let t be a sensor deployment strategy such that $\mathcal{L}^0 \cap \mathcal{K}(t) = \emptyset$. If $\rho^{\widetilde{x}} \geq \rho^0$ for a state $\widetilde{x} \in \mathcal{P}$ generated under level l, the contribution to $Z^{\mathcal{K}}$ of all states $\widetilde{y} \in \mathcal{Y}^2 \Leftrightarrow s^k(\widetilde{y}) \geq s^k(\widetilde{x})\ \forall k \in \mathcal{K}$ is given by equation 5.8, and the states in $\mathcal{P}$ which require further evaluation belong to one of $2l$ possible partitions $\mathcal{P}_v(d, \widetilde{x})$ defined by equations 5.10 to 5.13 for every value of $1 \leq d \leq l$.

$$\alpha\left(\mathcal{X}^I(d,\widetilde{x})\right) = \left\{s^1(\widetilde{x}), s^2(\widetilde{x}), \ldots, s^{d-1}(\widetilde{x}), a^d(\mathcal{X}), \ldots, a^K(\mathcal{X})\right\} \tag{5.10}$$

$$\beta\left(\mathcal{X}^I(d,\widetilde{x})\right) = \left\{b^1(\mathcal{X}), b^2(\mathcal{X}), ..., b^{d-1}(\mathcal{X}), s^d(\widetilde{x}) - 1, b^{d+1}(\mathcal{X}), ..., b^K(\mathcal{X})\right\} \tag{5.11}$$

$$\alpha\left(\mathcal{X}^{II}(d,\widetilde{x})\right) = \left\{s^1(\widetilde{x}), s^2(\widetilde{x}), ..., s^{d-1}(\widetilde{x}), a^d(\mathcal{X}), ..., a^K(\mathcal{X})\right\} \tag{5.12}$$

$$\beta\left(\mathcal{X}^{II}(d,\widetilde{x})\right) = \left\{s^1(\widetilde{x}), s^2(\widetilde{x}), ..., s^{d-1}(\widetilde{x}), s^d(\widetilde{x}) - 1, b^{d+1}(\mathcal{X}), ..., b^K(\mathcal{X})\right\} \tag{5.13}$$

$$r(\mathcal{X}) = \prod_{k \in \mathcal{K}} \left(\sum_{t=s^k(\widetilde{x})}^{t=b^k(\mathcal{X})} p_t^k \right) \tag{5.14}$$

Notice that in order to avoid duplicating a state evaluation, the first state to be considered in partitions $\mathcal{X}^I(d, \widetilde{x})$ is such that $s^{d-1} = a^{d-1} + 1$. Additionally, notice that depending on the characteristics of the set $\mathcal{P}$ it may not be possible to generate $\mathcal{X}^I(d, \widetilde{x})$ and $II^2(d, \widetilde{x})$. This is the case when $s^d(\widetilde{x}) = a^d$, or when $s^j = S^j\ \forall\ j : 1 < j < d - 1$.

Figure 5.3 represents Partitioning Rule 1 within the tree-representation context. It is easy to see that the implementation of this rule is equivalent to pruning the state space trees by eliminating the children of all the nodes at the l^{th} level of set $\mathcal{P}$ for which $w > s^l$. It follows a description of the algorithmic details of the implementation of this rule.

Algorithmic implementation The algorithmic implementation will be described based on the tree-pruning representation of the state partitioning process, which has a more intuitive interpretation. Figure 5.4 summarizes the

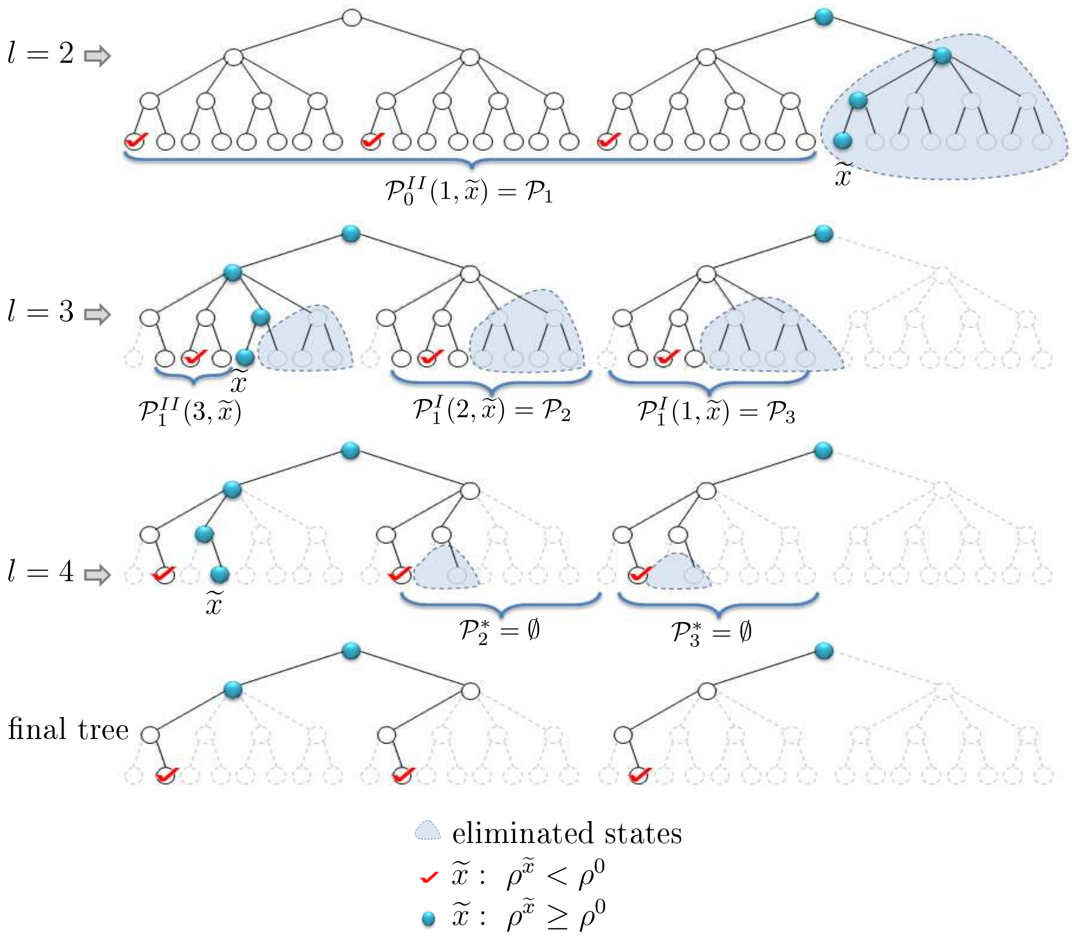

Figure 5.3: Tree-pruning interpretation of Partitioning Rule I

procedure, exemplified in Figure 5.5. The algorithm works by levels, and it maintains two sets of partitions, $\mathcal{Q}(l)$ and $\mathcal{R}(l)$, which contain all the states of $\mathcal{P}$ that may require evaluation. The partitions in $\mathcal{Q}(l)$ are active, in the sense that the corresponding states, based on level l, still need to be generated and evaluated if appropriate. The set $\mathcal{R}(l)$ encompasses partitions for which the $l-$based states have already been generated, thus do not to be considered at the current level. At every iteration, the first partition $\mathcal{P}_0 \in \mathcal{Q}(l)$ is selected, and the corresponding states generated and evaluated according to algorithm 1. Based on Partitioning Rule 1, when a state $\widetilde{x} \in \mathcal{P}_0 : \rho^{\widetilde{x}} \geq \rho^0$ is found, the algorithm performs a partitioning operation, which includes:

1. Identifying the subsets of states $\widetilde{y}$: $s^k(\widetilde{y}) \geq s^k(\widetilde{x}) \; \forall \; k \in \mathcal{K}$ and computing their contribution to $Z^{\mathcal{K}}$ (equation 5.14)

2. Creating partitions $\mathcal{P}_v \in \mathcal{Q}(l)$ as defined by Equations 5.10 and 5.11

3. Updating the remaining partitions in $\mathcal{Q}(l)$ using Equations 5.15 and 5.16, and adjusting the system expected cost (Equations 5.17 through 5.19)

4. Generating partitions in $\mathcal{R}(l)$ using Equations 5.12 and 5.13

5. Removing $\mathcal{P}_0$ from $\mathcal{Q}(l)$

If the last state $\widetilde{x} \in \mathcal{P}_0$ is reached without partitioning the set, $\mathcal{P}_0$ is moved to $\mathcal{R}(l)$, and after all the partitions in the active set are processed the algorithm moves to the next level, and $\mathcal{R}(l)$ becomes the active set. The procedure ends when both sets of partitions are empty. The third step in the process is essentially equivalent to step 2, but it involves additional verifications, given that the condition $\rho^{\widetilde{x}} > \rho^0$ does not necessarily have an impact in all the existing partitions in $\mathcal{Q}(l)$. Equations 5.15 and 5.16 describe the updated partitions.

$$\alpha(\mathcal{P}_v^*(d)) = \{aux^1(\widetilde{x}), aux^2(\widetilde{x}), ..., aux^{d-1}(\widetilde{x}), a^d(\mathcal{P}_0), ..., a^K(\mathcal{P}_0) \tag{5.15}$$

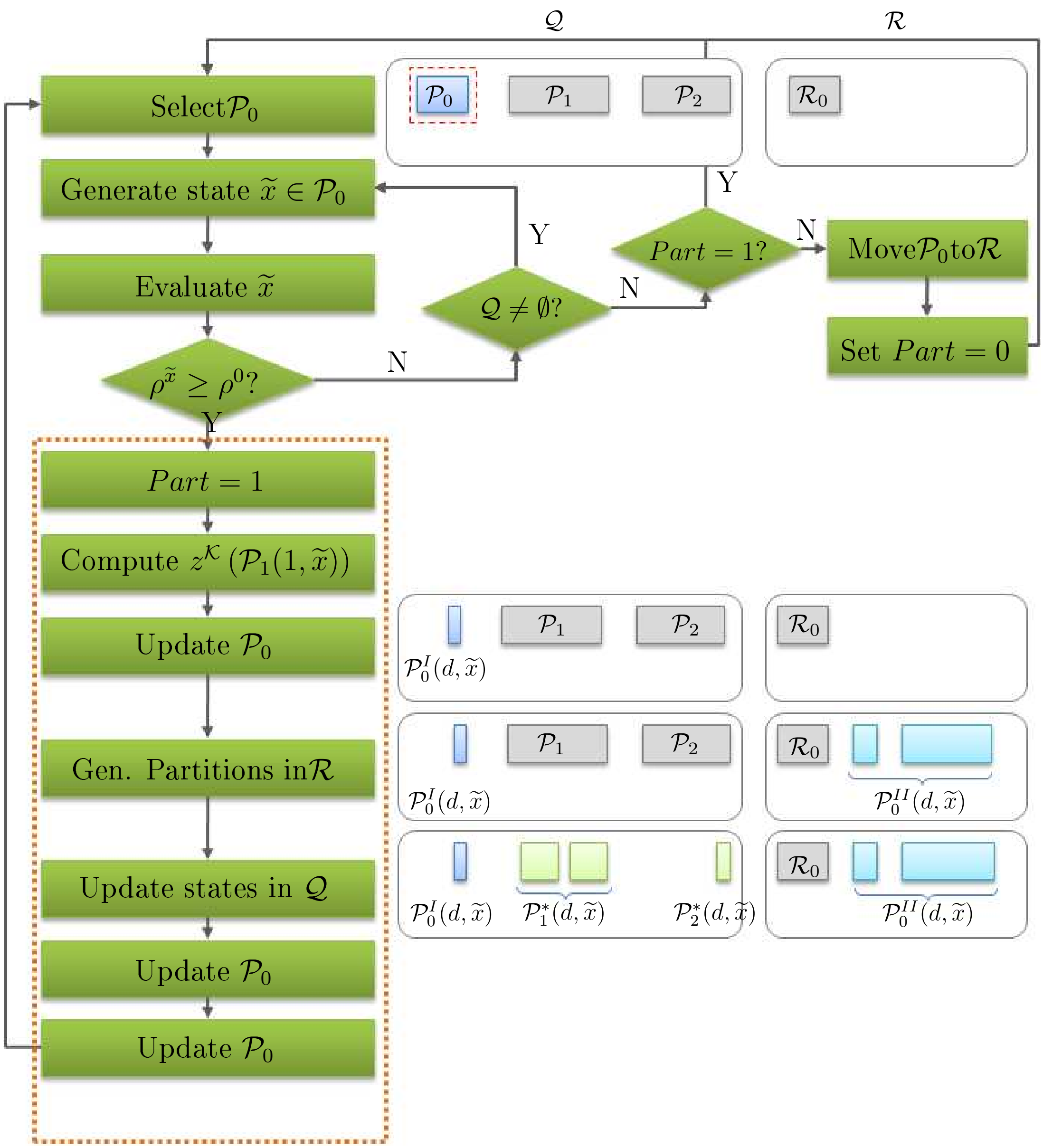

Figure 5.4: Evaluation subroutine for strategies Type I

$$\beta(\mathcal{P}_v^*(d)) = \left\{b^1(\mathcal{P}_0), b^2(\mathcal{P}_0), ..., b^{d-1}(\mathcal{P}_0), aux^d(\widetilde{x}) - 1, b^{d+1}(\mathcal{P}_0), ..., b^K(\mathcal{P}_0)\right\} \tag{5.16}$$

The auxiliary array aux is computed according to algorithm 2, and only those partitions such that $b^k(\mathcal{P}_v) \geq a^k(\mathcal{P}_v)\ \forall k \in \mathcal{K}$ are modified.

Algorithm 2 Definition of aux in state-space partitioning algorithm

for all $(k \in \mathcal{K})$ **do**
  if $(s^k(\widetilde{x}) > b^k(\mathcal{P}_0))$ **then**
    $aux^k = b^k(\mathcal{P}_0)$
  else
    $aux^k = s^k(\widetilde{x})$
  if $(s^k(\widetilde{x}) < a^k(\mathcal{P}_0))$ **then**
    $aux^k = a^k(\mathcal{P}_0)$
  else
    $aux^k = s^k(\widetilde{x})$

If a partition is updated, the system expected cost is adjusted using Equations 5.17 through 5.19.

$$r(\mathcal{P}_v^*(d)) = \prod_{k \in \mathcal{K}} \left(\sum_{g=aux^k}^{g=b^k(\mathcal{P}_v)} p_v^k \right) \tag{5.17}$$

$$z^{\mathcal{K}}(\mathcal{P}_v^*(d)) = \rho^0 \times r(\mathcal{P}_v^*(d)) \tag{5.18}$$

$$Z^{\mathcal{K}} = Z^{\mathcal{K}} + z^{\mathcal{K}}(\mathcal{P}_v^*(d)) \tag{5.19}$$

5.1.3.2 Strategies Type II

The partitioning rule for deployment strategies Type II presents the same structure as Partitioning Rule 1. However, for this case $\mathcal{L}^0 \supseteq \mathcal{K}$, and therefore ρ^0 can no longer be considered an upper bound on the shortest path cost under different perceived network states. An alternate bound is given by the shortest path cost on a network from which all links $k \in \mathcal{K}$ are removed. Let τ and $\mathcal{L}^{\overline{\mathcal{K}}}$ denote such cost and the corresponding path, respectively. Based on this

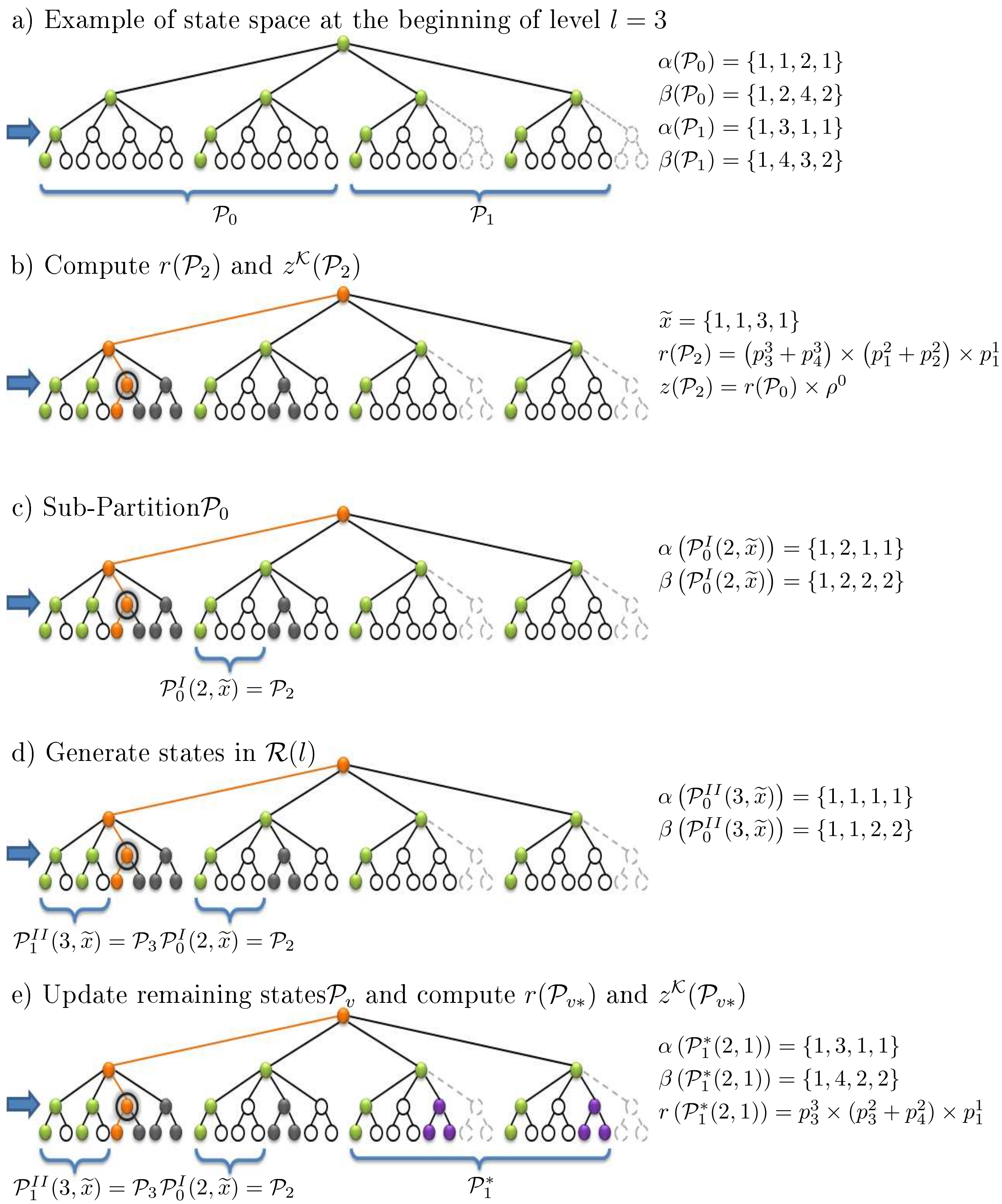

Figure 5.5: Basic state space partitioning procedure when $\rho^{\widetilde{x}} = \rho^0$

new condition and on shortest path properties, the following facts are used to derive an appropriate partitioning rule for strategies Type II:

- **Fact 3:** For states $\widetilde{x} \in \mathcal{X}$ such that $c^k(\widetilde{x}) < \mu^k \ \forall\ k \in \mathcal{K}$, the shortest path cost $\rho^{\widetilde{x}}$ is given by the cost perceived at $\mathcal{L}^0$.

Proof: Assume that the default optimal cost is ρ^0 on path $\mathcal{L}^0$, and that the minimum expected cost under state $\widetilde{x}$ occurs on path $L' \neq \mathcal{L}^0$, such that $c_{\widetilde{x}}(L') < c_{\widetilde{x}}(\mathcal{L}^0)$. Given that costs change only for links in $\mathcal{K}$, the costs under $\widetilde{x}$ may be re written as $c_{\widetilde{x}}(L') = c_0(L') + \sum_{k\in\mathcal{K}}(c^k_{\widetilde{x}} - \mu^k)$ (Assuming that $\mathcal{K} \subseteq L'$, which provides a lower bound on the value of $c_{\widetilde{x}}(L')$) and $c_{\widetilde{x}}(\mathcal{L}^0) = \rho^0 + \sum_{k\in\mathcal{K}}(c^k_{\widetilde{x}} - \mu^k)$. In view of our starting assumption, $c_0(L') + \sum_{k\in\mathcal{K}}(c^k_{\widetilde{x}} - \mu^k) < \rho^0 + \sum_{k\in\mathcal{K}}(c^k_{\widetilde{x}} - \mu^k)$. The former implies that $c_0(L') < \rho^0$, which contradicts the assumption the ρ^0 is optimal and completes the proof.

Corollary: If a state $\widetilde{x} \in \mathcal{X}$ is such that $c^k(\widetilde{x}) \leq \mu^k \ \forall\ k \in \mathcal{K}$, the shortest path cost is defined by Equations 5.22, which does not require an additional shortest path evaluation.

$$\rho^{\widetilde{x}} = \rho^0 - \sum_{k\in\mathcal{K}} \left(\mu^k - c^k(\widetilde{x})\right) \tag{5.20}$$

- **Fact 4:** If $\rho^{\widetilde{x}} = \tau$ for a perceived network state $\widetilde{x} \in \mathcal{X}$, then $\rho^{\widetilde{y}} = \tau \ \forall\ \widetilde{y} :$ $s^k(\widetilde{y}) \geq s^k(\widetilde{x}) \ \forall k \in K$. The proof of this fact is equivalent to that of Fact 2.

Corollary: If the shortest path value under a perceived network state is $\rho^{\widetilde{x}} = \tau$, equations 5.21 and 5.9 can be used to compute the contribution to $Z^{\mathcal{K}}$ of all states $\widetilde{y} \in \mathcal{Y}^2$ such that $s^k(\widetilde{y}) \geq s^k(\widetilde{x}) \ \forall k \in \mathcal{K}$.

$$z^{\mathcal{K}}(\mathcal{Y}^2) = \tau \times r(\mathcal{Y}^2) \tag{5.21}$$

In view of the former, we define the following partitioning rule and corresponding algorithmic implementation:

Partitioning Rule 2

> Let $\mathcal{X}$ be a K-dimensional space describing the perceived network state space based on the information provided by K sensors, or a subset of such space. Let t be a sensor deployment strategy such that $\mathcal{L}^0 \supseteq \mathcal{K}(t)$, and let τ be the value of the shortest path on a network from which all links $k \in \mathcal{K}(t)$ are removed. If $\rho^{\widetilde{x}} = \tau$ for a state $\widetilde{x} \in \mathcal{X}$ generated under level l, the contribution to $Z^{\mathcal{K}}$ of all states $\widetilde{y} \in \mathcal{Y}^2 \Leftrightarrow s^k(\widetilde{y}) \geq s^k(\widetilde{x}) \ \forall \ k \in \mathcal{K}$ is given by equations 5.21 and 5.9. The states in $\mathcal{X}$ which require further evaluation belong to one of $2l$ possible partitions $\mathcal{P}_v$ defined by equations 5.10 to 5.13.

The intuitive interpretation of this strategy is similar to the one provided for Partition Rule 1. Notice that, depending on the characteristics of the network, it may not be possible to find τ. When this is the case, the state-space cannot be partitioned based on the shortest path value. However, some problem properties can be utilized to reduce the number of shortest path evaluations, described in Section 5.1.3.4.

Algorithmic Implementation The implementation of Partition Rule 2 is almost identical to the one described in Section 5.1.3.1, the only difference being the conditions which trigger a partitioning operation. In addition to state-space partitioning, the subroutine used to evaluate deployment strategies of Type II takes advantage of Fact 3 to reduce the number of shortest path evaluations. Algorithm 3 summarizes the approach.

5.1.3.3 Strategies Type III

The partitioning rule adopted for deployment strategies Type III is identical to that one described for strategies of Type II. The strategy evaluation subroutine, described by 4 also makes use of the following fact:

- **Fact 4:** For states $\widetilde{x} \in \mathcal{X}$ such that $c^k(\widetilde{x}) \leq \mu^k \ \forall \ k \in \mathcal{K} \cap \mathcal{L}^0$ and

Algorithm 3 Evaluation Subroutine for Strategies Type II

for $(1 < l < K)$ **do**
  for all $(P_v \in \mathcal{Q}(l))$ **do**
    while (Algorithm 1 returns state $\widetilde{x}$) **do**
      Compute $r(\widetilde{x})$ (Equation 5.2)
      if $(s^k(\widetilde{x}) \leq \varphi^k \ \forall\ k \in \mathcal{K})$ **then**
        $\rho^{\widetilde{x}} = \rho^0 - \sum_{k \in \mathcal{K}} \left(\mu^k - c^k(\widetilde{x})\right)$
        $z^{\mathcal{K}}(\widetilde{x}) = \rho^{\widetilde{x}} \times r(\widetilde{x})$
      else
        Run shortest path and compute $\rho^{\widetilde{x}}$
        if $(\rho^{\widetilde{x}} = \tau)$ **then**
          Run partition subroutine (Figure 5.4)
          Move to next P_v
        else
          $z^{\mathcal{K}}(\widetilde{x}) = \rho^{\widetilde{x}} \times r(\widetilde{x})$

$c^k(\widetilde{x}) \geq \mu^k \ \forall\ k \in \mathcal{K} \wedge k \notin \mathcal{L}^0$the shortest path cost $\rho^{\widetilde{x}}$ is given by the cost perceived at $\mathcal{L}^0$.

Proof: Assume that the default optimal cost is ρ^0 on path $\mathcal{L}^0$, and that the mini mu expected cost under state $\widetilde{x}$ occurs on path $L' \neq \mathcal{L}^0$, such that $c_{\widetilde{x}}(L') < c_{\widetilde{x}}(\mathcal{L}^0)$. Given that costs change only for links in $\mathcal{K}$, the costs under $\widetilde{x}$ may be re written as $c_{\widetilde{x}}(L') = c_0(L') + \sum_{k \in \mathcal{K} \cap \mathcal{L}^0}(c^k_{\widetilde{x}} - \mu^k) + \sum_{k \in \overline{\mathcal{K} \cap \mathcal{L}^0}}(c^k_{\widetilde{x}} - \mu^k)$ (Assuming that $(\mathcal{K} \cap \mathcal{L}^0) \subseteq L'$, which provides a lower bound on the value of $c_{\widetilde{x}}(L')$) and $c_{\widetilde{x}}(\mathcal{L}^0) = \rho^0 + \sum_{k \in \mathcal{K} \cap \mathcal{L}^0}(c^k_{\widetilde{x}} - \mu^k)$. In virtue of our starting assumption, $c_0(L') + \sum_{k \in \overline{\mathcal{K} \cap \mathcal{L}^0}}(c^k_{\widetilde{x}} - \mu^k) < \rho^0$. Notice that the second term in the right hand side in this equation is non-negative under the conditions specified for Fact 4, in view of which $c_0(L') < \rho^0$, which contradicts the assumption regarding the optimality of ρ^0 and completes the proof.

Corollary: If a state $\widetilde{x} \in \mathcal{X}$ is such that $c^k(\widetilde{x}) \leq \mu^k \ \forall\ k \in \mathcal{K} \cap \mathcal{L}^0$ and $c^k(\widetilde{x}) \geq \mu^k \ \forall\ k \in \mathcal{K} \wedge k \notin \mathcal{L}^0$, the shortest path cost is defined by equation 5.22, which does not require an additional shortest path evaluation.

$$\rho^{\widetilde{x}} = \rho^0 - \sum_{k \in \mathcal{K} \cap \mathcal{L}^0} \left(\mu^k - c^k(\widetilde{x})\right) \tag{5.22}$$

Algorithm 4 Evaluation Subroutine for Strategies Type III

for $(1 < l < K)$ **do**
  for all $(P_v \in \mathcal{Q}(l))$ **do**
    while (Algorithm 1 returns state $\widetilde{x}$) **do**
      Compute $r(\widetilde{x})$ (Equation 5.2)
      if $(s^k(\widetilde{x}) \leq \varphi^k \; \forall \; k \in \mathcal{K} \cap \mathcal{L}^0$ and $s^k(\widetilde{x}) \geq \varphi^k \; \forall \; k \in \mathcal{K} \wedge k \notin \mathcal{L}^0)$ **then**
        $\rho^{\widetilde{x}} = \rho^0 - \sum_{k \in \mathcal{K}} \left(\varphi^k - c^k(\widetilde{x})\right)$
        $z^{\mathcal{K}}(\widetilde{x}) = \rho^{\widetilde{x}} \times r(\widetilde{x})$
      else
        Run shortest path and compute $\rho^{\widetilde{x}}$
        if $(\rho^{\widetilde{x}} = \tau)$ **then**
          Run partition subroutine (Figure 5.4)
          Move to next P_v
        else
          $z^{\mathcal{K}}(\widetilde{x}) = \rho^{\widetilde{x}} \times r(\widetilde{x})$

5.1.3.4 Partitioning rule for temporary state space subdivision

The partition rules described above are equivalent to a tree-pruning operation which permanently removes a portion of the state-space trees. The following fact allows identifying sections of the tree which may be temporarily disregarded during the analysis of a particular level, even though the corresponding nodes cannot be eliminated from the tree.

- **Fact 5:** Define $\mathcal{L}^{\widetilde{x}}$as the shortest path under state $\widetilde{x} \in \mathcal{X}$, and let $w \in \mathcal{K}$ be any of the equipped links under strategy t. If $w \notin \mathcal{L}^{\widetilde{x}}$ then for all states $\widetilde{y} \in \mathcal{Y}^3 \iff c^w(\widetilde{y}) \geq c^w(\widetilde{x}) \wedge c^k(\widetilde{y}) = c^k(\widetilde{x}) \; \forall k \in \mathcal{K}, \; k \neq w$ it is true that $w \notin L^{\widetilde{y}}$.

Proof: Let link w connect nodes i and j, and F_{s-i} and F_{j-t} be the optimal sub paths connecting the origin to node i, and node j to the destination under state $\widetilde{x}$. If $w \notin \mathcal{L}^{\widetilde{x}}$, $c_{\widetilde{x}}(F_{s-i}) + c^w_{\widetilde{x}} + c_{\widetilde{x}}(F_{j-t}) > \rho^{\widetilde{x}}$. Under the conditions established by Fact 5, $\rho^{\widetilde{x}}$ is the cost of a feasible path for all states in $\mathcal{Y}^3$, and the left hand side of the last equation, which represents the cost of the shortest path using link ij, remains larger than $\rho^{\widetilde{x}}$. In view of the later, no path using link w may be the shortest under the states contained in $\mathcal{Y}_3$, which completes the proof.

Corollary: If under state $\widetilde{x} \in \mathcal{P}_v$, $\mathcal{P}_v \in \mathcal{Q}(l)$ link $w \in \mathcal{K}$ is found not to belong to the shortest path, the corresponding shortest path value $\rho^{\widetilde{x}}$ can be extrapolated to all states $\widetilde{y} \in \mathcal{Y}^3 \iff s^k(\widetilde{y}) = s^k(\widetilde{x})\ \forall k \in \mathcal{K},\ k \neq w$, and $s^w(\widetilde{y}) \geq s^w(\widetilde{x})$. The contribution of such states to the total system expected cost under strategy t is given by Equations 5.23 and 5.24.

$$z^{\mathcal{K}}(\mathcal{Y}^3) = \rho^{\widetilde{x}} \times r(\mathcal{Y}^3) \tag{5.23}$$

$$r(\mathcal{Y}^3) = \left(\prod_{k \in K, k \neq w} p^k(\widetilde{x})\right) \times \left(\sum_{s=s^w(\widetilde{x})}^{s=b^w(P_v)} p^w_{s^k}\right) \tag{5.24}$$

Fact 5 may be used as the basis for a number of different partitioning rules. The one presented below was developed taking into account practical implementation considerations. The criterion utilized to design this rule was to take advantage of Fact 5 in order to reduce the number of shortest path computations without greatly increasing the data storage requirements, or the complexity of the necessary data structures. The rule was also developed to fit within the framework imposed by the selected state generation procedure (Algorithm 1).

Partition Rule 4:

> Let $\mathcal{P}_v \subseteq \mathcal{Q}(l)$ be a K-dimensional space describing a partition of the state space defined by the information provided by K sensors $k \in \mathcal{K}$. Assume that states are generated according to algorithm 1. If during the evaluation of state $\widetilde{x} \in \mathcal{P}_v$ $\exists w \in \mathcal{K} :\ w < l,\ w \notin L^{\widetilde{x}}$, then all states $\widetilde{y} : s^k(\widetilde{y}) = s^k(\widetilde{x})\ \forall k \in \mathcal{K},\ k \neq w$ and $s^w(\widetilde{y}) \geq s^w(\widetilde{x})$ can be evaluated in a single step according to 5.23 and 5.24. This leads to a subdivision of the remaining states in $\mathcal{Q}(l)$ into at most 3 sets $\mathcal{T}_v(\widetilde{x}, w, l)$, such that $\alpha\left(\mathcal{T}_v(\widetilde{x}, w, l)\right) = \left\{a^k(\mathcal{P}_v)\right\}$, and $\beta\left(\mathcal{T}_v(\widetilde{x}, w, l)\right)$ is defined according to Figure 5.6. The first state

to be considered in each of these partitions is $\widetilde{x}^0$, described in the same figure.

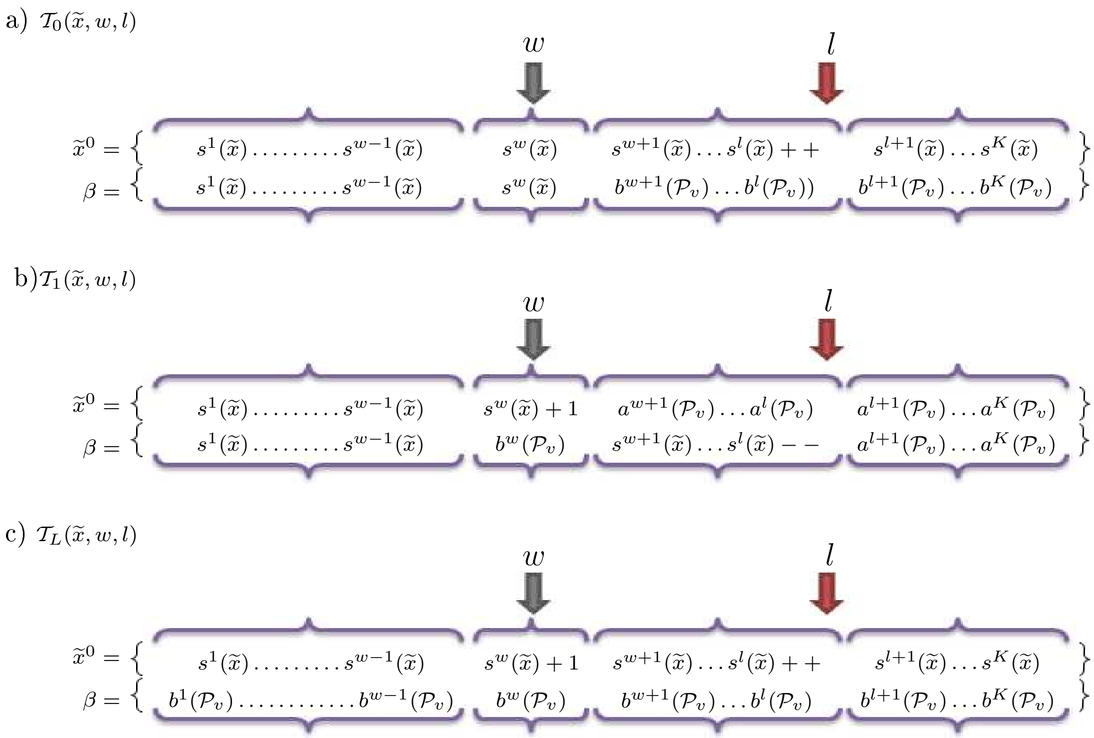

Figure 5.6: Temporary sub partitions based on $\widetilde{x}$ when $\exists\, w \in \mathcal{K}:\ w \notin L^{\widetilde{x}}$

In Figure 5.6 expressions $s^l --$ and $s^l ++$ are used to identify the indices corresponding to the states immediately before and immediately after $\widetilde{x}$ (according to the state generation process presented in Algorithm 1), for the range $w+1 < k < l$. Notice that such states may not exist , in which case the corresponding partition is not created. Partitions $\mathcal{T}_v(\widetilde{x}, w, l)$ are inserted into $\mathcal{Q}(l)$ in the position previously occupied by $\mathcal{P}_v$ in the order in which they are generated.

The algorithmic implementation can accommodate cases on which more than one link $w_i \in \mathcal{W}:\ \mathcal{W} \subseteq \mathcal{K}$ is not part of the shortest path under $\widetilde{x}$. The corresponding probabilities are given by Equation 5.25.

$$r^{adj}(\mathcal{T}_v(\widetilde{x}, w_i, l)) = \sum_{s=s^{w_1}(\widetilde{x})}^{s=b^{w_1}(\mathcal{P}_v)} \left(p_s^{w_i} \times \prod_{k\in\mathcal{K},\ k\neq w_i} p^k_{s^k(\widetilde{x})} \right) + \sum_{w_j\in\mathcal{W},\ j>i} \left\{ \left(\sum_{s=s^w(\widetilde{x})+1}^{s=b^w(\mathcal{P}_v)} p_s^{w_j} \right) \times \prod_{k\in\mathcal{K},\ k\neq w_i} p^k_{s^k(\widetilde{x})} \right\} \tag{5.25}$$

The implementation also accounts for the fact that during the evaluation of the states in $\mathcal{T}_v(\widetilde{x}, w, l)$ it is possible to find new states $\widetilde{x}' : w \notin L^{\widetilde{x}'}$. Let $\mathcal{W} = \{w_1, w_2, ..., w_W\}$ be the subset of all links in $\mathcal{K}$ which do not belong to the shortest path, and assume that the sub indices preserve the link ordering followed when defining the tree levels. Define $\widetilde{x}^*(w_j) = \left\{s^{w_j+1}, s^{w_j+2}, ..., s^l\right\}$ as the portion of the array $\widetilde{x}$ which contains the link state indices corresponding to $k : \ w_j + 1 \leq k \leq l$, and notice that this segment is enough to generate all partitions $\mathcal{T}_v(\widetilde{x}, w_j, l)$. The proposed algorithm maintains a list of such arrays for each level $1 \leq d < l$, denoted $X^*(w_j)$, and utilizes it to create temporary partitions $\mathcal{T}_v(\widetilde{x}^*(w_j), w_j, l)$ when s^{w_j} is increased by algorithm 1. For notational simplicity, well denote the elements in $X^*(w_k)$ using $\widetilde{x}^*_i$ whenever w_j can be inferred from the context.

We will denote $k^{last}(\widetilde{x})$ the index increased in order to generate $\widetilde{x}$. Lists $X^*(w_j)$ are reset every time a new state is generated for all $l > j > k^{last}$.

Each time a shortest path evaluation is accomplished, the algorithm identifies all the links which are not part of the shortest path, extrapolates the shortest path value to the corresponding states, and saves $x^*(w_j)$ for all links $j : \ \ j < l$ by inserting it into the appropriate position in $X^*(w_j)$. The state generation algorithm is slightly modified with respect its original version in order to verify if the set $x^*(k^{last}(\widetilde{x}))$ is empty for every new state $\widetilde{x}$. If this is not the case, temporary sub partitions are generated, which skip those states already evaluated based on the fact that $k^{last}(\widetilde{x})$ was not part of the shortest path. Each of the temporary partitions is processed following the standard procedure, with a slight modification in the implementation of Partitioning Rule 4. Within a temporary partition $\mathcal{T}_v(\widetilde{x}, w_j, l)$ only those links links k such that $k \leq w_j$ are considered for the application of Partitioning

Rule 4. This is a heuristic simplification, seeking to reduce the amount of information that needs to be stored in order to keep track of the states already evaluated. Figure 5.7 illustrates the underlying concept within the context of a tree representation of the state-space. The corresponding algorithmic implementation (Algorithm 5) acts as the framework in which the remaining subroutines are inserted. Section 5.1.4 presents the pseudo code and discuses some implementation considerations. Numerical analyses on two test network are presented in Section 5.1.4.

5.1.4 State-space partitioning algorithm: Summary

Algorithm 5 describes the implementation of Partitioning Rules 1 through 4 to the evaluation of sensor deployment strategies. The procedure is slightly different depending on whether or not the strategy involves placing sensors on links belonging to the shortest path under no information, $\mathcal{L}^0$. Strategies Type I are such that none of the monitored links belongs to $\mathcal{L}^0$, while strategies Type II and III place all or some of the sensors on $\mathcal{L}^0$, respectively. For all strategies the algorithm starts working on a state space which includes all possible cost realizations on the measured links. These realizations define the perceived network states, and lead to different shortest expected cost values $\rho^{\widetilde{x}}$. The contribution to the expected shortest path cost of each realization is given by $\rho^{\widetilde{x}} \times r(\widetilde{x})$, where $r(\widetilde{x})$ is the corresponding probability. Under some circumstances the same value of $\rho^{\widetilde{x}}$ can be applied to a number of perceived states, which leads to a subdivision of the state space. For strategies Type I this is possible for states $\widetilde{x} : \ \rho^{\widetilde{x}} = \rho^0$. Under strategies Type II and III, the state-space is partitioned if $\rho^{\widetilde{x}} = \tau$, where τ is the value of the shortest path on a network from which all the equipped links are removed. The algorithm also takes advantage of other problem properties in order to reduce the number of shortest path computations (Sections 5.1.3.2, 5.1.3.3 and 5.1.3.4).

The state space can be visualized as a set of trees (Section 5.1.2), and the state generation procedure used in this algorithm is equivalent to growing the trees from top to bottom, generating at every level (associated with a particular

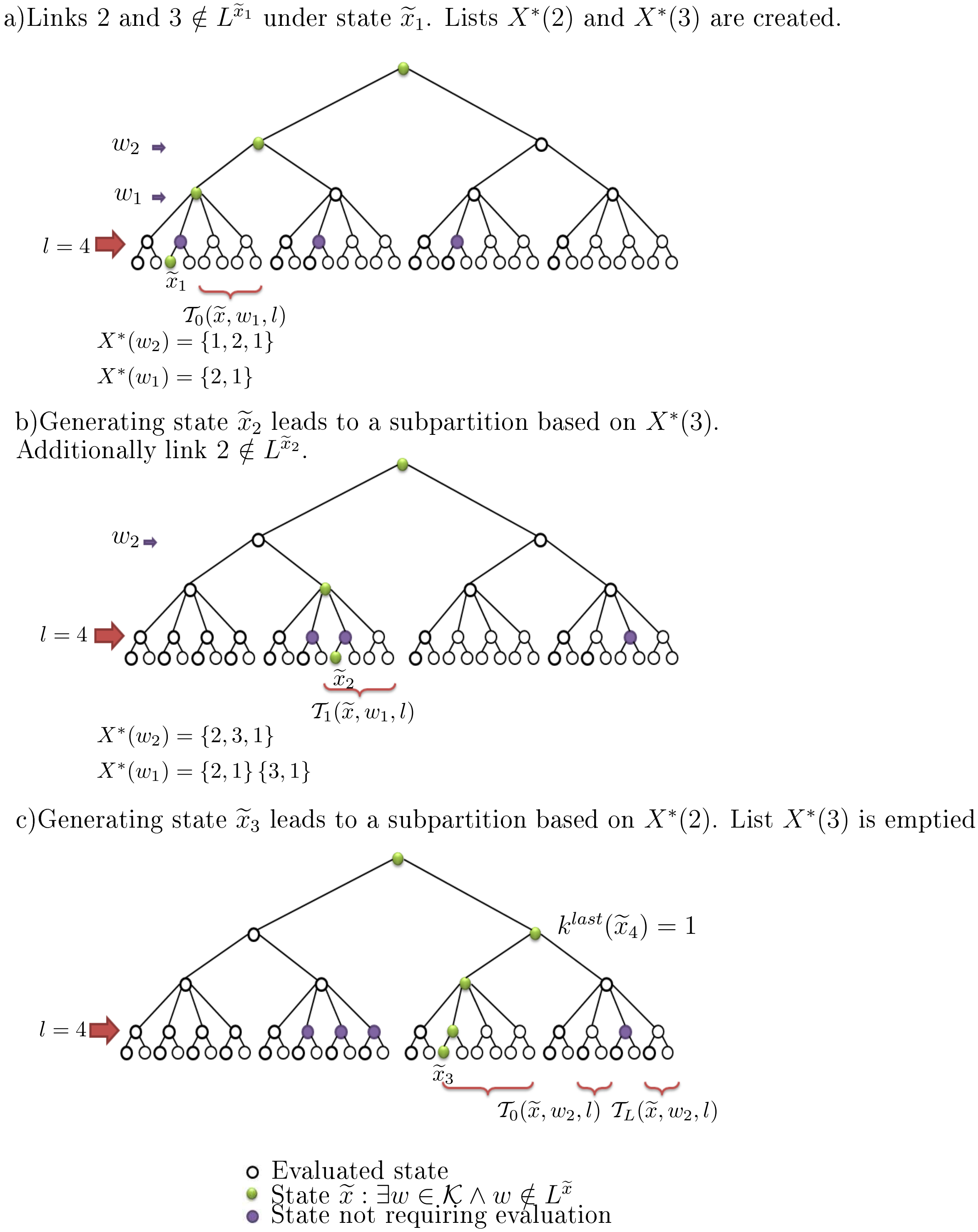

Figure 5.7: Partitioning Rule 4

Algorithm 5 State Space Partitioning Algorithm

$\alpha^0 = (1, 1, ...1)$
$\beta^0 = (S^1, S^2, ..., S^K)$
if t is of Type I **then**
  $Ref = \rho^0$
else
  $Ref = \tau$
for all $(l \in \mathcal{K})$ **do**
  while $(\mathcal{Q}(l) \neq \emptyset)$ **do**
    while (Algorithm 1 returns state $\widetilde{x}$) **do**
      if evaluation is necessary (Algorithm 3 or 4) **then**
        Find $\mathcal{L}^{\widetilde{x}}$ and $\rho^{\widetilde{x}}$
        Compute $r(\widetilde{x})$
        if $(\rho^{\widetilde{x}} < Ref)$ **then**
          if $(X^*(k^{last}(\widetilde{x})) \neq \emptyset \ \& \ flag \geq k^{last})$ (if temporary partitions are necessary) **then**
            for all $(\widetilde{x}^* \in X^*(k^{last}(\widetilde{x})))$ **do**
              Generate $\mathcal{T}_j(\widetilde{x}^*, k^{last}(\widetilde{x}), l)$ and insert them in $\mathcal{Q}(l)$
            Generate $\mathcal{T}_2(\widetilde{x}^*, k^{last}(\widetilde{x}), l)$ insert it at the top of $\mathcal{Q}(l)$
            Generate $\mathcal{T}_3(\widetilde{x}^*, k^{last}(\widetilde{x}), l)$ and insert it in$\mathcal{Q}(l)$
            $flag = k^{last}(\widetilde{x})$
            $X^*(k^{last}(\widetilde{x})) = \emptyset$
            Move to the first element $\widetilde{x} \in \mathcal{Q}(l)$
          else
            for all $(w : \ w \in \mathcal{K} : w < k^{last}, w \notin L^{\widetilde{x}})$ **do**
              Store $\widetilde{x}^*(w)$ in $X^*(w)$
            Compute $r^{adj}(\widetilde{x})$ (equation 5.25)
            $z(\widetilde{x}) = r^{adj}(\widetilde{x}) \times \rho^{\widetilde{x}}$
        else
          Partition (Algorithms 3, 4, and Figure 5.4)
          Update $\mathcal{Q}(l)$ and $\mathcal{R}(l)$
          Compute $z(\widetilde{x})$
          Move to the first element of $\mathcal{Q}(l)$
      else
        Move to next $\widetilde{x}$ in $\mathcal{P}_v$
  $\mathcal{Q}(l) = \mathcal{R}(l)$
  $\mathcal{R}(l) = \emptyset$

link) a branch for each possible cost realization. The tree configuration varies depending on the order on which the links are sorted (i.e. which link is assigned to each level). Even though the total number of leaves (and therefore states to be evaluated) is constant, the effectiveness of the partitioning procedure clearly depends on the tree structure. The numerical experiments presented in 5.1.5 suggest that the methodology is more efficient when links are considered in increasing order of their number of states, resolving ties based on links cost range $\Delta\varepsilon^j$ (Equation 5.26). These results are consistent with the theoretical approach presented in Alexopoulos [1997]. Additionally, for strategies Type III it is advantageous to assign the links in $\mathcal{L}^0 \bigcap \mathcal{K}$ to the highest levels (closer to the top).

$$\Delta\varepsilon^j = \varepsilon^j_{S^j} - \varepsilon^j_1 \tag{5.26}$$

The algorithm was designed to evaluate sensor deployment strategies when the underlying objective function involves routing assets between one origin and one destination. It can be utilized to assess cases on which more than one origin-destination (O-D) pair exist, by defining $\rho^{\widetilde{x}}_{ALL} = \sum_{od\in\mathcal{O}} \rho^{\widetilde{x}}_{od} \times h_{od}$, where $\mathcal{O}$is the set of all the considered O-D pairs, and $h_{od}$is an optional variable used to assign different weights to the O-D pairs. In this case $L^{\widetilde{x}}_{od}$ is replaced by $T^{\widetilde{x}} : \ j \in T^{\widetilde{x}} \Leftrightarrow \exists\, od \in \mathcal{O} : \ j \in L^{\widetilde{x}}_{od}$. Even though the worst case complexity for this algorithm is exponential (all the states may be generated and evaluated) the results displayed below suggest a much better performance in practice.

5.1.5 Numerical testing

The algorithm described in Section 5.1.4 was tested on one of the example networks presented in Alexopoulos [1997]. Example Network 1 has 10 nodes and 23 links which costs are described by discrete probability distributions with 2 to 5 states (Table 5.2 and Figure5.8). The algorithm was implemented in C++, and two different versions were developed. The first one utilizes the data structures described in Section 5.1, and it incorporates Partitioning Rules 1 through 3, according to Algorithm 3. This version was utilized for most of

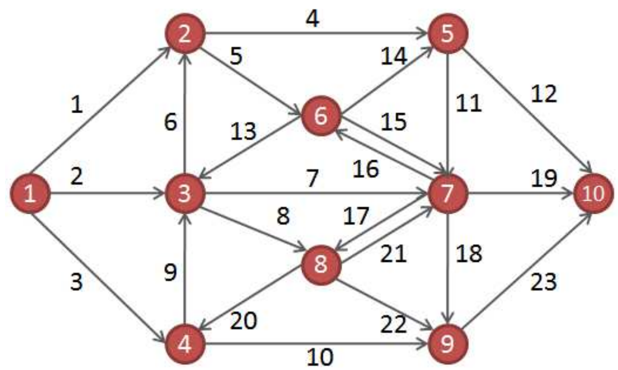

Figure 5.8: Network I topology

the numerical tests included in this work. The second version, denoted "tree implementation", explicitly utilizes the tree representation described in Section 5.1.2. Trees are stored as lists of nodes, and additional arrays are used to indicate the starting node of each level, and to implement the state partitions. Even though the tree version allows for an easier implementation of complex partitioning rules, such as partitioning rule 4, it often requires data structures of the size of the state-space, which limits its applicability.

In order to test the state space partitioning algorithm, all possible sensor deployment strategies of size n=[1, 5] were evaluated in Network I. The large number of such strategies (C_n^N) allowed to assess the algorithm performance, which was measured in terms of the reduction in the number of shortest path computations required to evaluate a sensor deployment plan. The running time was not used as an indicator of performance given that the software implementation was not designed focusing on computational efficiency. Furthermore, different approaches were taken to implement naïve approaches and state-space partitioning methodologies, seeking to explore available coding resources such as the Boost Graph Library. As a consequence of this approach, the running times are likely to underestimate the potential of the state-space partitioning methodologies. Future research will design software tools allowing for a valid running time comparison.

We denote N_{eval}^t the number of network states for which a shortest path

Link	ε_1^j	p_1^j	ε_2^j	p_2^j	ε_3^j	p_3^j	ε_4^j	p_4^j	ε_5^j	p_5^j
1	70	*0.2*	73	*0.5*	94	*0.3*				
2	25	*0.5*	35	*0.4*	82	*0.1*				
3	42	*0.2*	48	*0.3*	61	*0.5*				
4	26	*0.1*	31	*0.2*	55	*0.4*	88	*0.2*	90	*0.1*
5	58	*0.3*	70	*0.3*	95	*0.4*				
6	15	*0.4*	73	*0.6*						
7	65	*0.4*	74	*0.5*	75	*0.1*				
8	59	*0.6*	72	*0.3*	98	*0.1*				
9	21	*0.3*	32	*0.2*	85	*0.3*	98	*0.2*		
10	89	*0.7*	96	*0.3*						
11	32	*0.2*	48	*0.2*	67	*0.6*				
12	63	*0.5*	99	*0.5*						
13	66	*0.8*	85	*0.1*	98	*0.1*				
14	6	*0.1*	15	*0.4*	39	*0.3*	58	*0.2*		
15	2	*0.4*	48	*0.6*						
16	61	*0.2*	63	*0.3*	85	*0.5*				
17	16	*0.2*	18	*0.3*	40	*0.3*	52	*0.2*		
18	3	*0.1*	30	*0.4*	50	*0.5*				
19	16	*0.1*	34	*0.5*	71	*0.4*				
20	90	*0.5*	96	*0.5*						
21	21	*0.3*	46	*0.4*	85	*0.3*				
22	17	*0.1*	49	*0.4*	53	*0.4*	65	*0.1*		
23	6	*0.1*	12	*0.1*	54	*0.3*	66	*0.5*		

Table 5.2: Link cost probability distribution for Network I

computation was performed in order to evaluate strategy t. The total number of states generated by the algorithm is given by N^t_{eval}, and N^t_{max} represents the number of possible states under strategy t. The algorithm savings are measured by the reduction in the total number of shortest path evaluations $\Delta N^t = N^t_{max} - N^t_{eval}$, or its percent expression $\Delta N^t = N^t_{max} - N^t_{eval}$. The number of partitions generated during an evaluation is $\#P$. Partitions are stored using two arrays of size n, and we define $\#P_{sim}$ as the maximum number of partitions simultaneously stored during an evaluation. The sub partitions defined in Section 5.1.3.4 are also stored as sets of two arrays of varying size, and are included in the computation of $\#P_{sim}$. A small number of simultaneous partitions is desirable in order to reduce memory requirements (the actual number of shortest path evaluations is not a linear function of the performed partitions). Additionally, handling the sub partitions, particularly the ones generated based on Rule 4, can be computationally expensive, slowing down the overall process. The later justifies the selection of relatively simple partitioning rules for the present application (5.1.3.4). Nevertheless, for problems such that the evaluation of a single state is relatively complex, e.g. network assignment, additional benefits could be derived from refined sub partitioning schemes.

Tables 5.3 and 5.4 summarize the observed algorithmic performance for the basic implementation (Including Partitioning Rules I through III). In this table $\overline{N_{eval}}$ is the average value of N^t_{eval} across all the strategies t involving the same number of sensors. Similarly, $\overline{N_{eval}\%}$ denotes the average value when the number of shortest path evaluations is expressed as a percentage of the maximum number of states $\overline{N_{eval}\%} = \frac{1}{T}\sum_{t\in T}\frac{N^t_{eval}}{N^t_{max}}$.

On Network I less than 20% of all possible states were evaluated for strategies of Type I, and an even smaller percentage of the possible states required evaluations for K=3 and K=4 (9% and 8% respectively). The corresponding results for strategies of Type II and III are less impressive, which is likely to be a consequence of the lack of alternative paths connecting the selected origin and destination in when the measured links are removed. As a result, the bound on the shortest path cost, τ, cannot be computed,

	2 Sensors			3 Sensors		
	Type III	Type I	Type II	Type III	Type I	Type II
#	60	190	3	630	1440	1
$\overline{N^t_{max}}$	9.3	9.57	9	28.64	29.45	27
$\overline{N^t_{gen}}$	8.75	1.34	8.67	26.8	2.7	25.0
$\overline{N^t_{eval}\%}$	73	15	88	84.1	8.6	88.9
$\#P$	1.00	0.10	1.00	1.11	0.21	2.00

Table 5.3: Algorithmic performance in Network I (a)(Basic implementation)

	4 Sensors		6 Sensors	
	Type III	Type I	Type III	Type I
#	4010	4845	62187	38760
$\overline{N^t_{max}}$	88.03	90.28	822.96	838.39
$\overline{N^t_{gen}}$	88.03	90.28	791.99	106.30
$\overline{N^t_{eval}\%}$	89.87%	7.70%	95.20%	10.50%
$\#P$	1.16	0.38	1.25	0.98

Table 5.4: Algorithmic performance in Network I (b) (Basic implementation)

limiting the applicability of Partitioning Rules 1 through 3. When used to analyze a larger network (Network II, described in the following section), the basic implementation exhibited a much better performance on strategies of Type II and III (Table 5.5) than on Network I. This suggests that the scarceness of alternative paths is only a concern when the number of deployed sensors is a high percentage of the total number of links, or in networks with poor connectivity. Furthermore, Partitioning Rule 4 can be used to mitigate the effect of such conditions.

Tables 5.6 and 5.7 present the results obtained by applying the tree version, which includes partitioning rule 4, to the analysis of sensor deployment strategies in Network I and II, respectively. Figure 5.9 compares these results with the ones presented in Tables 5.3 and 5.5.

The tree implementation was observed to deliver a better performance than the basic version described above, particularly for the evaluation of strategies of Type III in Network I. For Network II, the performance of both

	2 Sensors			3 Sensors		
	Type III	Type I	Type II	Type III	Type I	Type II
#	117	741	3	2340	9139	1
$\overline{N^t_{max}}$	7.18	9.46	5.33	21.78	29.03	12
$\overline{N^t_{gen}}$	4.43	1.19	4.00	12.48	1.94	6
$\overline{N^t_{eval}\%}$	39%	13%	75%	45%	7%	0.5
$\#P$	1.00	0.06	1.00	1.17	0.14	2

Table 5.5: Algorithmic performance in Network II (Basic implementation)

	$\overline{N_{eval}\%}$		
	Type III	Type I	Type II
2	57.7%	14.1%	81.5%
3	34.1%	6.6%	85.2%
4	17.1%	3.3%	81.1%
5	9.6%	1.8%	
6	6.2%	1.2%	

Table 5.6: Algorithmic performance in Network I (Tree implementation)

	$\overline{N_{eval}\%}$		
	Type III	Type I	Type II
2	32.2%	14.1%	32.4%
3	23.4%	5.5%	41.7%
4	11.0%	2.5%	35.0%

Table 5.7: Algorithmic performance in Network II (Tree implementation)

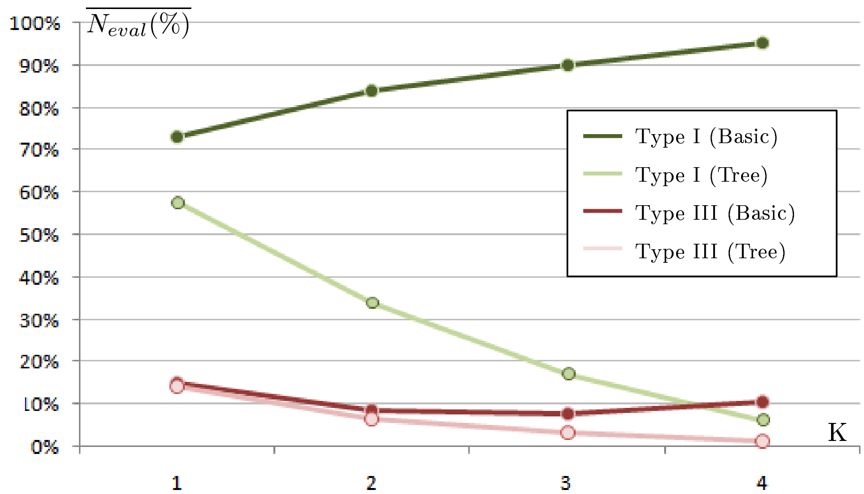

Figure 5.9: Comparison of partitioning strategies

implementations is comparable across all strategies.

The presented results also suggest that even though the total number of shortest path computations increases with K, the algorithm performance remains stable in terms of the percentage of evaluated states, except on those cases for which the total number of sensors affects network connectivity.

Table 5.9 illustrates the impact of the order in which in $\mathcal{K}$ are considered on the algorithm performance for strategies of size 5 on Network I. The up (down) arrows in this table are used to indicate that the corresponding quantity increases (decreases) for increasing values of l. The best performance was achieved when links j were sorted in increasing order of S^k, breaking ties based on an ascending value of $\Delta\varepsilon^j$ (equation 5.26). This sorting strategy presents the advantage of considering links with a smaller range of possible costs at early stages. Such links are more likely to lay outside (or inside) the shortest path regardless of their value, which is used as the basis to prune the state-space trees. By performing the pruning operations at higher levels (lower value of l), the number of states to be considered in further levels is reduced. Similarly, given that the number of tree branches initialized at each level l is a function of S_k for all $k < l$, it is desirable to postpone the consideration of links with a large number of states to the later stages of the algorithm, in order

Criteria	$\overline{\Delta S}$ (%)
$\uparrow \Delta\varepsilon^j$	5.21
$\downarrow \Delta\varepsilon^j$	6.11
$\uparrow S^j$	5.17
$\downarrow S^j$	5.96
$\uparrow \Delta\varepsilon^j \uparrow S^j$	5.21
$\uparrow S^j \uparrow \Delta\varepsilon^j$	5.08
Random	5.39

Table 5.8: Impact of the link-sorting criteria on the algorithmic performance (Network I, $K = 5$)

to take advantage of previous pruning and avoid unnecessary evaluations. It is interesting to notice that the algorithm performance does not deteriorate greatly when the links are considered in a random order, which may be a desirable alternative for specific implementations, including those involving a very large number of sensors.

Additional experiments were conducted to assess the performance of the methodology for different link cost probability distributions functions: A uniform distribution (Table A.3) was constructed by assigning equal probabilities to all states described on table 5.2. Two strongly asymmetric distributions were generated by which assigning 70% of the probabilities to the most (Right-skewed) and least (Left-skewed) expensive cost realizations (Tables A.4 and A.5 respectively). The remaining variations (Tables A.6 and A.7) analyze the impact of the total number of states on the algorithmic performance. The results suggest that the methodology is slightly sensitive to the characteristics of the probability distribution, but consistently leads to a very significant reduction of the total number of shortest path computations. Uniform distributions, and those tilted to the right, require a larger number of computations than others, given that they present more link states which are below the average, thus may require a shortest path evaluation.

Distribution	$\overline{N_{eval}}(\%)$	$\overline{N_{eval}}(\%)$ TI	$\overline{N_{eval}}(\%)$ TIII
Uniform	5.3	2.09	8.92
Left-skewed	4.66	1.65	7.15
Right-skewed	6.35	3.98	8.21
Two States	17.99	1.82	9.6
Five States	1.68	0.92	2.28

Table 5.9: Impact of the link probability distribution on the algorithmic Performance (Network I, $K = 5$)

5.1.6 Summary

This section introduced a state-space partitioning algorithm which reduces the number of shortest path computations required to evaluate the performance of different sensor deployment strategies on the expected cost of an adaptive-system optimum assignment procedure. The technique takes advantage of problem characteristics to develop partitioning rules and evaluation criteria. These are used to find the shortest path value corresponding to several states based on a single evaluation, or on previously computed values. Most of these rules are rooted on shortest path properties, and incorporate some concepts of shortest path re-optimization. They involve comparing the shortest path value under a given state to a threshold value, and also verifying whether the measured links are part of the optimal solution under a particular cost realization. Unlike the cases studied by Alexopoulos [1997], who introduced some of the seminal concepts implemented in this section, the computation of the system expected cost under an adaptive system optimum paradigm entails knowing the exact value of the shortest path for every perceived state, increasing the number of required evaluations. The procedure is specialized for three different sensor deployment strategy types in order to exploit the problem characteristics. Strategies are distinguished based on whether or not some, all or none of the monitored links belong to the shortest path under no information, $\mathcal{L}^0$. Such condition affects the selection of the above mentioned threshold value, leading to slightly different partitioning rules and evaluation criteria. Section 5.1.2 presents a way to visualize the state space generation procedure as a tree growing process, and the corresponding partitioning rules as tree-pruning operations. The proposed representation is useful to understand how the methodology works, and can be utilized in the design of more complex partitioning rules in future extensions.

Section 5.1.5 analyzes the performance of two different implementation of the developed algorithm in C++. Both sets of results exhibit major reductions in the total number shortest path evaluations required to assess the system performance resulting for a specific sensor deployment plan. On average, strategies were appraised by computing the shortest paths corresponding to only 10% of the possible states. The performance varied across strategy types, and the proportion of evaluated states ranged between 2% and 35% in a well connected network. For poorly connected networks, or in cases involving a high number of deployed sensors relative to the network size, evaluations of up to 85% of the possible states were necessary. The later is a consequence of the impossibility of computing the threshold value necessary to partition the state-space, but it does not indicate a deterioration of the procedure performance as a function of the state space size. Furthermore, the conducted experiments suggest a stable performance for increasingly large state spaces. The results also indicate that the, even though the algorithm is affected by the characteristics of the link cost probability distribution, the observed impacts are not major. Distributions assigning lower probabilities to states with low costs were found to require a larger number of evaluations, as a consequence of the increased number of link states with cost below the corresponding expected value.

The methodology presented in this section is effective in reducing the computational effort involved in evaluation the expected cost of adaptive system optimum strategies under different sensor deployment patterns. This is a critical in the search for optimal sensor deployment strategies, given that the proposed methodology may involves evaluating a very large number of possible strategies. The next section describes a Tabu heuristic approach designed to reduce the number of sensor deployment plans to be considered.

5.2 Selecting an optimal sensor deployment strategy: a Tabu heuristic approach

Given the integer, non-convex nature of the models introduced in Chapter 4, finding an optimal sensor deployment strategy may entail evaluating every possible solution. Even though the procedure presented in Section 5.1 reduces the computational burden of each of these evaluations, the number of possible strategies grows rapidly with the network size and the number of sensors, thus finding the exact solution may easily become prohibitive. The Tabu search procedure described in this section provides an alternative solution approach which, despite not guaranteeing the optimality of the solution, was found to provide very good results in practice in considerably low computational times.

Tabu search is a popular meta heuristic technique for the solution of combinatorial optimization problems (Glover [1977]). Extensions and refinements to the original approach have been used to solve a variety of problems, including integer and mixed integer programs (F.Glover [1989]). The methodology is clearly suitable for the solution of the problem addressed in this chapter, which can be modeled as a binary quadratic program (Chapter 4).

Tabu search typically starts form a feasible solution (trial solution), and selects a "move" s which transforms the existing solution t into a new solution ($s(t) = t'$) by changing the value of one or more of the problem variables. A move can be defined as a mapping on a subset of the solution space. The fundamental concept underlying Tabu search is that, by dynamically choosing subsets of forbidden (Tabu) moves, one may generate a search process which balances intensification and diversification, thoroughly exploring the solution space without falling into local optima. The set of allowed moves is defined as the neighborhood of a trial solution. Banned moves are stored in one (or more) Tabu lists, which are dynamically updated. An evaluation function is utilized to select the "best" move in the set of feasible moves, which is typically the one leading to the greatest improvement (or the least disimprovement) of the objective function. An additional advantage of Tabu search is that its implementation can be tailored to the characteristics of the problem under study by changing the way in which the Tabu list is created and managed, the definition of moves, and the specification of the evaluation function, among others (see F.Glover [1990] for examples).

In order to find the sensor deployment strategy which optimizes the performance of adaptive system optimum routing strategies we implement a Tabu search heuristic which utilizes adaptive memory structures similar to the ones proposed by Glover et al. [1998] for the solution of binary quadratic problems. These store the recency and frequency information used to guide the solution search, basically keeping track of the values assigned to variables in recent iterations. Adaptive memory structures have been proved to be very effective in the solution of problems classified as hard in the literature (Glover et al. [1998], Pardalos and Rodgers [1990]). Unlike the problems analyzed in Glover et al. [1998], the selection of an optimal sensor deployment strategy is constrained, given the fixed number of sensors which need to be deployed. This is reflected in the definition of "moves" presented in the following section. Additionally, some of the concepts presented by Ahuja et al. [2002] in their study of local search algorithms for very large neighborhoods were implemented in the search for a more efficient performance. The following sections describe the proposed Tabu search implementation (Section 5.2.1), and the corresponding numerical tests (Section 5.2.2).

5.2.1 Algorithm description

Let Γ be the K-dimensional solution space containing all feasible sensor deployment strategies t involving placing K sensors on links $k \in \mathcal{K}(t)$, and denote $Z^{\mathcal{K}}(t)$ the corresponding system expected cost under the information provided by such sensors (the notation $Z(\mathcal{K})$ is occasionally used in this discussion). Our problem can be stated in a simplified manner using equation 5.27

$$Min\ Z^{\mathcal{K}}(t): \quad t \in \Gamma \tag{5.27}$$

where the value of $Z^{\mathcal{K}}(t)$ can be appraised using the methodology described in Section 5.1. Our decision variables, t, consist of binary M-tuples $t = \{g^t(1), g^t(2).....g^t(M)\}$, where $g(j) = 1$ if $j \in \mathcal{K}(t)$and $g(j) = 0$ otherwise, and $M = |\mathcal{M}|$ is the cardinality of the set of network links.

We define a f-distance **move** $m^f(t, t')$ between trial solutions t and t' as the swap of f elements currently in $\mathcal{K}(t)$ for elements k'currently not in $\mathcal{K}(t)$. A swap is accomplished by setting $g^{t'}(k) = 0$ for the exiting elements, and $g^{t'}(k') = 1$ for the elements "entering" the solution. Notice that this definition of move always maintains a feasible problem solution, and that the cardinality of $\mathcal{K}(t)$ remains constant. The swap moves are implemented in a compounded fashion (Ahuja et al. [2002], Congram et al. [2002]), which selects entering and exiting links independently from each other. Links exiting $\mathcal{K}(t)$ are also chosen based on their individual impact on the objective function.

The value of f is the "depth" of the move, and the proposed algorithm implements a variable depth search scheme (Ahuja et al. [2002]), under which f changes cyclically between an upper and a lower bound (u and l, respectively), in an attempt to balance intensification and diversification. Small values of f permit a thorough search within a specific region of the solution space. Large values of the variable are used to escape from local optima. Each value of f defines a mode η_i, characterized by the corresponding depth f^{η_i} and a span value $n(f^{\eta_i})$, which represents the number of moves of depth f^{μ_i} to be performed. The total number of modes, I, as well as the number of cycles through such modes, C, are problem parameters. Algorithm 8, which summarizes the Tabu procedure, also describes the cyclic variations of f based on critical events, defined below.

Two adaptive **memory structures** (or Tabu lists) per link are used to guide the search process: a recency list R^j with elements $R^j[l]$, and a frequency list F^j, both of which are updated after a "critical event" is encountered (Algorithm 6). A critical event is a move $m^f(t, t')$ such that $Z^{\mathcal{K}}(t') > Z^{\mathcal{K}}(t)$ (i.e. causes a deterioration in in the objective function value). The Tabu lists keep track of the elements in $\mathcal{K}(t)$ which are part of the solution before such event, and therefore can be considered part of a local optima. The recency list has a finite length of sr which reflects the desired short memory span. It is managed as a circular list, in such way that elements are added to the bottom of the list and removed from its top. At any iteration T, equation 5.28 can be used to find Tabu_R(j), the number of times that a link j has been part of a critical solution in the last sr moves. Even though the size of R is typically adjusted heuristically, practical experiments (Glover et al. [1998]) suggest that values between 3 and 12 lead to efficient implementations for a variety of applications.

$$\text{Tabu\_R}(j) = \sum_{d=T-sr}^{d=T} g^d(j) = \sum_{l=1}^{l=sr} R^j[l] \tag{5.28}$$

The frequency list (Tabu_F(j)) records the participation of links j in critical solutions throughout the execution of the algorithm. The frequency list can be stored as an M-dimensional array which elements reflect the number of times a link has been part of a critical solution since the beginning of the algorithm (Equation 5.29).

$$\text{F\_count}(j) = \sum_{d=1}^{d=T} g^d(j) = \sum_{l=1}^{l=sr} F^j[l] \tag{5.29}$$

Algorithm 6 Updating the Tabu Lists at Critical Events

for all $j \in \mathcal{M}$ **do**
  Push back $g^j(t)$ into R^j
  $\text{R\_count}(j) = \text{R\_count}(j) + g^j(t) - R^j[1]$
  $\text{F\_count}(j) = \text{F\_count}(j) + g^j(t)$
  Remove $R^j[1]$

At each iteration an **optimal move** is identified, which entails selecting the links leaving and entering the solution in such way that the objective function decreases the most (or increases the least). This is accomplished by evaluating $Z^{\mathcal{K}}(t')$ for each possible move, and selecting the strategy rendering the lowest objective function value.

Within the decomposed swap movement framework implemented here, a move $m^f(t,t')$ is accomplished in two independent steps, denoted DELETE and ADD. At both stages, evaluations are conducted in order to assess the impacts of the considered action on the objective function. The Tabu lists are used to adjust the results of the corresponding evaluations through factors which reduce the attractiveness of links identified as part of local optimal solutions. Such links are therefore less likely to be reincorporated to (or maintained in) the considered solution in the short term, which fosters diversification in the search process.

The links to be deleted are selected individually from $\mathcal{K}(t)$ based on the impact that their removal has on the current objective function value. The later is given by Δ_k(equation 5.30), which is computed accounting for recency and frequency information. The links exhibiting the f lowest values of Δ_k are removed from $\mathcal{K}(t)$, and the resulting subset is denoted $\mathcal{K}'(t) = \mathcal{K}(t) - (k_1, k_2....k_f)$. . A random factor R (discussed below) is used to randomize the selection process within certain limits, given that the impact measured by $\Delta_k$does not exactly reflect the aggregate impact of removing a subset of links.

$$\Delta_k = Z^{\mathcal{K}'}(t) - \mathrm{z}_R(k) - \mathrm{z}_F(k) \tag{5.30}$$

$$z_R(k) = \text{Tabu\_R}(k) \times a \tag{5.31}$$

$$z_F(k) = \text{Tabu\_F}(k) \times b \tag{5.32}$$

In Equations 5.31 and 5.32, a and b are **penalty factors**, heuristically determined. For this application, $a = \max_j(\delta_{max}(j))$, where $\delta_{max}(j)$(equation 5.33) represents the maximum possible reduction in the total system expected cost introduced by placing a sensor on link j, and is an upper bound on the benefits of obtaining information about the state of link j (Section 4.4). The value of b is defined as a function of the number of iterations, following the experience described in Alexopoulos [1997]. For this application $b = \frac{1}{10^3 \times \ iter}$.

$$\delta_{max}(j) = \sum_{s:\ \varepsilon_s^j < \varphi^j} \left(\varphi^j - \varepsilon_s^j\right) \times p_s^j \tag{5.33}$$

During the ADD stage, f links $j \in \mathcal{K}'(t)$ are chosen simultaneously to enter the new solution, based on the improvement they cause on the objective function. The **neighborhood** of a solution t, $N(t)$, is therefore defined as the set of all possible f-tuples formed by links currently not in $\mathcal{K}(t)$. Such neighborhood may be very large, and the proposed implementation evaluates only a subset of $N(t)$, a fairly common practice in local search algorithms for very large neighborhoods (Ahuja et al. [2002]). In order to generate the reduced neighborhood, candidate links are sorted based on their $\delta_{max}^T(j)$ value (adjusted to account for recency and frequency information according to Equation 5.34), and Algorithm 7 is used to systematically generate a percentage G of the possible f-tuples (or combinations).

$$\delta_{max}^T(j) = \delta_{max}(j)\mathrm{z}_R(k) + \mathrm{z}_F(k) \tag{5.34}$$

This percentage varies between a lower bound G^l, which guarantees a minimum exploration of the neighborhood, and an upper bound G^u, selected in order to avoid an excessive computational burden. The actual size of the reduced neighborhood G can be smaller than its upper bound if an objective function value lower than the prevalent optimal solution is found before performing G^u evaluations.

The procedure designed to generate the reduced neighborhood (Algorithm 7), despite incorporating a random element, is systematic, in such way that if $G = 100\%$ all the possible combinations are generated without repetition.

In the corresponding algorithm, $A = \frac{1}{G^l}$ defines the number of combinations to "skip" (based on the lexicographic order described earlier) if only G^l $f-$tuples were to be generated. Array $v[i]$ contains the link indices corresponding to the generated f-tuple, and array $\max[i]$ and $\min[i]$ represent the maximum and minimum value that the index my adopt at each position.

Algorithm 7 Generation of a randomized reduced f- swap neighborhood

```
a = max[f] − v[f]
v[f] = max[f] − (A − a) − 1
r_count=0
v → aux_v
while (a<A) do
  j = f
  while (v[j] = max[j]) do
    j − −
  if (j≥0) then
    v[j] + +
    for (j + 1 ≤ k ≤ f) do
      v[k] = v[k − 1] + 1
  else if (r_count<A) then
    r_count++
    aux_v → v
    while (v[j] = max[j]) do
      j − −
    v[j] + +
    for (j + 1 ≤ k ≤ f) do
      v[k] = v[k − 1] + 1
    a = A

for (0 ≤ i ≤ f) do
  v[i] = v[i] + R
```

Possible Combinations			
v[1]	v[2]	v[3]	Order
1	2	3	***1***
1	2	4	***4***
1	2	5	***7***
1	3	4	
1	3	5	***2***
1	4	5	***5***
2	3	4	
2	3	5	
2	4	5	***3***
3	4	5	***6***

Figure 5.10: Generation of a reduced neighborhood

These arrays are initialized setting $\min[i] = i$ and $\max[i] = \overline{|\mathcal{K}(t)|} - (f - i)$. R is a random number, different for every ADD operation, used to avoid considering the same reduced neighborhood in all iterations. The maximum and minimum values of R (R^u and R^l) are problem parameters. Figure 5.10depicts the order in which 6 combinations would be generated for a move of depth 3 on a network with $|\mathcal{K}'| = 5$ and $G^l = 30\%$, when 7 neighborhood evaluations are desired.

During the ADD operation, f-tuples in the reduced neighborhood are generated and added to $\mathcal{K}'(t)$, generating a k-tuple $\mathcal{K}''(t)$ which is evaluated using the methodology described in the previous section. The corresponding objective function value is adjusted according to equation 5.35, and the neighborhood member $v[i]$ leading to the lowest value of Δ_v defines the optimal move for the corresponding iteration.

$$\Delta_v = Z^{\mathcal{K}''}(t) + \sum_{i=0}^{i=f} (\mathrm{z}_R(v[i]) + \mathrm{z}_F(v[i])) \tag{5.35}$$

Algorithm 8 describes the integration of all the previously described elements into a Tabu Search heuristic, which parameters were tested and adjusted through the numerical experiments described in the following section.

Algorithm 8 Managing the search depth and span

$Z^* = Z(0)$, iter=0
while (cycle<C) **do**
  $i = 1$, dir=1
  while (move_count$\leq n(f^{\eta_i})$) **do**
    iter++
    Perform move $m^{f(\eta_i)}(t, t')$
    if $(Z(t') < Z^*)$ **then**
      $Z^* = Z(t)$
      move_count++
    else if $(Z(t') > Z(t))$ **then**
      Critical_Event=1
      Update Tabu_F
      if (dir=1) **then**
        move_count=$n(f^{\mu_i})$
    else
      move_count++
    Update Tabu_R
  if (dir=1) **then**
    $i++$
  else
    $i--$
  if $(i = I)$ **then**
    $dir = -1$
  else if $(i = 1)$ **then**
    dir = 1
  if $(i = 1)$ **then**
    cycle++

5.2.2 Numerical testing

The tests performed in this section are conducted on two medium-sized networks, Network I (previously introduced in Figure 5.8 and Table 5.2), and Network 2 (Figure A.1 and Table A.2). The later accounts for 15 nodes and 42 arcs, and it was also utilized in Alexopoulos [1997].

The tests measured the performance of the Tabu methodology in terms of the number of strategy evaluations conducted before reaching an optimal solution (Section 5.2.2.2). Test results were also used to adjust the parameters described in Section 5.2.1, as described in Section 5.2.2.1. In order to assess the algorithmic performance, optimal solutions were obtained using the procedure described in Section 5.1 to evaluate all possible K sensors deployment strategies in Networks I and II. Due to practical considerations, the value of K ranged between 1 and 5 for Network I, and between 1 and 6 for Network II.

5.2.2.1 Parameter selection

This section discusses the selection of the various parameters utilized in the Tabu search heuristic introduced in Section 5.2.1. Most of the values were determined based on the literature, and adjusted to fit the requirements of the analyzed problem.

Appropriate values of I (the number of modes η_i), and the corresponding depths $f^{\eta_i}$and spans $n(f^{\eta_i})$, were identified by trial and error, and set to the values displayed in Table 5.10. These values have an impact on the number of evaluations conducted per cycle, and it was found that a large number of modes tends to delay the convergence process. In principle, it would suffice to utilize two modes, one with $f = 1$, aimed to intensify the search around near-optimal locations, and a second mode intended to diversify the search procedure by swapping most of the elements in the current solution. However, the problem properties discussed in Chapter 4.2 indicate that some of the beneficial impacts of information are attained only when specific sets of links are measured simultaneously. Swap movements of higher depths were included to facilitate a faster identification of such combinations. The number of moves within each mode was adjusted to avoid an excessive number of evaluations per cycle. The benefits of the Tabu search methodology are achieved only if there is a balanced distribution of intensification and diversification moves, which requires cycling between modes relatively fast.

K	f^{η_1}	$n(f^{\eta_1})$	f^{η_2}	$n(f^{\eta_2})$	f^{η_3}	$n(f^{\eta_3})$
1	1	1	-	-	-	-
2	1	5	2	1	-	-
3	1	5	3	1	-	-
4	1	5	4	1	-	-
5	1	5	3	2	5	1
6	1	5	4	2	6	1

Table 5.10: Parameters f^{η_i} and $n(f^{\eta_i})$

The maximum number of cycles, C, is set to 20 based on the results displayed in the following section (Table 5.11). An additional convergence criterion considers the number of moves accomplished since the last improvement in the objective function, and the total number of performed evaluations. For a wide variety of cases the algorithm was found to converge after a number of evaluations equivalent to 5% (or less) of the possible combinations. The additional convergence criterion terminates the program if the number of evaluated strategies is larger than 20% of the possible strategies, or if the solution has not improved in the last 50 moves. In the later case, $Y = \sum_{i=1}^{i=I} n(f^{\mu_i})$ moves of depth 1 are evaluated before terminating, based on some of the results discussed below.

The length of the short term memory structure, Tabu_R, was set to five iterations. On the networks considered in this study, lengths shorter than 3 iterations typically led to excessive cycling, and values higher than 8 delayed the convergence process. The results suggest that, even though the Tabu list should be long enough to guarantee that locally optimal variables do not reenter the solution within the same mode, allowing the reincorporation of such variables at the diversification stage of the same cycle may be beneficial for the convergence process.

Penalty values were chosen as described in the previous sections, and performed adequately. The parameter controlling the size of the reduced neighborhood, G, was assigned values between 50% and 80% of the $f-$neighborhood size. If the lower bound is reduced below 30%, the algorithm performance becomes more unstable. As a consequence, more evaluations are necessary to guarantee convergence, even though fewer strategies are evaluated per cycle.

The randomization parameter R is allowed to take values between 0 and $0.3 \times |\mathcal{K}'(t)|$, where $|\mathcal{K}'(t)|$ is the number of candidate links. This, combined with the sorting scheme used for the links in $\mathcal{K}'(t)$, gives a slightly higher priority to strategies including links with large values of $\delta_{max}(j)$.

Although most of the parameters were found to be adequate for various network sizes, some of them, such as G^u and R, may need to be adjusted for very large networks, in order to avoid an excessive number of evaluations. Similarly, the number an properties of swap modes is directly related to the problem characteristics, and should be adjusted appropriately.

5.2.2.2 Performance evaluation

The algorithm performance was measured in terms of the number of strategies evaluated before convergence, which represents the gains with respect to a naive approach under which all possible strategies of a given size need to be evaluated. Table 5.11 shows the results obtained on Network I for different numbers of deployed sensors. Given that the algorithm incorporates a random element, thirty runs were conducted for each case, in order to test the stability of the algorithm (the first row shows the average values across thirty runs conducted using different random seeds). The results, which suggest a stable algorithm performance for the selected values of R (Section 5.2.2.1), are very impressive for sensor deployment strategies of size 3 and larger. In most of these cases, the optimal value was found during the first 2 cycles ($C \leq 2$). Furthermore, the number of evaluated strategies E as a percentage of the total number of existing combinations E_{max} is very small in the majority of the numerical tests. It ranges from an average of 23% the in 3 sensor case, to only 3% for the 5 sensors case. For the 2 sensor case, the total number of strategy evaluations at convergence is equal, or even larger, than the evaluations performed under an optimal naive approach. The later reflects the fact that there is a minimum number of evaluations that the algorithm needs to perform in order to solve any problem. Such number is conditioned by C, mode parameters f^{η_i} and $n(f^{\eta_i})$, and the criteria used to define the neighborhood size. For large problems, the minimum required evaluations are a very small fraction of the solution space, and they have no impact on the algorithm performs. The solution space corresponding to the problem of finding the optimal sensor deployment strategy of size 2 in Network I is small, and the minimum number of computations performed by the proposed Tabu search procedure may exceed those involved in an optimal approach. This is not a concern, given that small problems can be easily solved using exact methodologies.

Figures 5.12 and 5.12 detail the convergence process as a function of the number of strategy evaluations for a representative model run in networks I and II respectively. In Figures 5.13 and 5.14 the same information is displayed, but the number of strategy evaluations is expressed as a percentage the possible number of strategies. In both sets of Figures, the error is given by equation 5.36

$$\text{Error} = \frac{Z^* - Z(t)}{Z^*} \cdot 100 \tag{5.36}$$

The first two figures suggest that the algorithm typically gets very close to the optimal solution in a relatively small number of iterations. For strategies involving a larger numbers of sensors, and for most strategies in the larger network, relatively long plateaus may be observed at low error values. These occur when the algorithm has identified most of the links in the optimal solution, but an insufficient number of intensification moves prevents it from finding the exact optimal solution. In order to improve convergence, the move depth f is set to 1 when a plateau is detected, regardless of the prevalent mode. The adjusted move depth is maintained during a number of moves

	$K=2$			$K=3$			$K=4$			$K=5$		
	C	E_{max}	$\frac{E}{E_{max}}$	C	E_{max}	$\frac{E}{E_{max}}$	C	E_{max}	$\frac{E}{E_{max}}$	C	E_{max}	$\frac{E}{E_{max}}$
A	0	351	1.39	1	380	0.21	0	288	0.03	2	910	0.03
1	1	521	2.06	0	185	0.16	0	255	0.03	1	404	0.01
2	0	133	0.53	1	280	0.44	0	280	0.03	0	262	0.01
3	0	113	0.45	3	785	0.20	1	303	0.03	1	387	0.01
4	0	106	0.42	1	350	0.28	1	423	0.05	10	2984	0.09
5	0	106	0.42	2	496	0.19	0	257	0.03	11	3650	0.11
6	0	196	0.77	1	329	0.14	0	256	0.03	1	334	0.01
7	1	497	1.96	0	253	0.20	0	280	0.03	1	482	0.01
8	1	423	1.67	1	351	0.17	1	329	0.04	6	1870	0.06
9	0	154	0.61	1	300	0.23	0	231	0.03	0	262	0.01
10	0	136	0.54	1	401	0.47	0	256	0.03	6	2062	0.06
11	0	157	0.62	3	831	0.28	0	297	0.03	0	266	0.01
12	1	399	1.58	1	492	0.12	0	233	0.03	1	434	0.01
13	2	606	2.40	0	209	0.16	0	281	0.03	0	308	0.01
14	2	863	3.41	1	281	0.32	1	305	0.03	3	1102	0.03
15	1	466	1.84	2	564	0.36	0	329	0.04	0	242	0.01
16	2	661	2.61	2	641	0.22	0	346	0.04	1	497	0.01
17	0	318	1.26	1	397	0.23	0	257	0.03	1	362	0.01
18	0	233	0.92	1	400	0.22	0	257	0.03	0	333	0.01
19	0	207	0.82	1	396	0.13	0	225	0.03	1	405	0.01
20	0	301	1.19	0	233	0.13	0	233	0.03	1	500	0.01
21	0	346	1.37	0	228	0.29	1	423	0.05	0	327	0.01
22	1	470	1.86	2	518	0.28	0	257	0.03	15	4606	0.14
23	0	129	0.51	2	493	0.12	1	327	0.04	1	382	0.01
24	0	207	0.82	0	209	0.14	0	273	0.03	3	1246	0.04
25	0	204	0.81	0	252	0.17	0	280	0.03	3	1124	0.03
26	2	808	3.19	1	305	0.14	0	255	0.03	1	411	0.01
27	1	584	2.31	1	256	0.12	0	257	0.03	0	261	0.01
28	2	617	2.44	0	209	0.23	0	351	0.04	0	238	0.01
29	0	305	1.21	1	415	0.19	0	321	0.04	4	1298	0.04
30	0	281	1.11	1	329	0.18	0	280	0.03	0	261	0.01

Table 5.11: Algorithm convergence in Network 1 for different random seeds

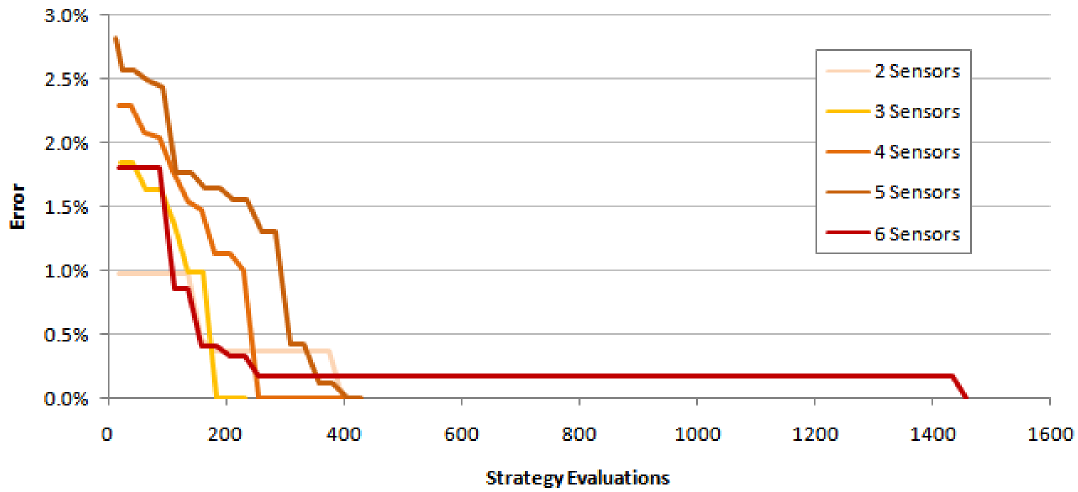

Figure 5.11: Algorithm convergence in Network I

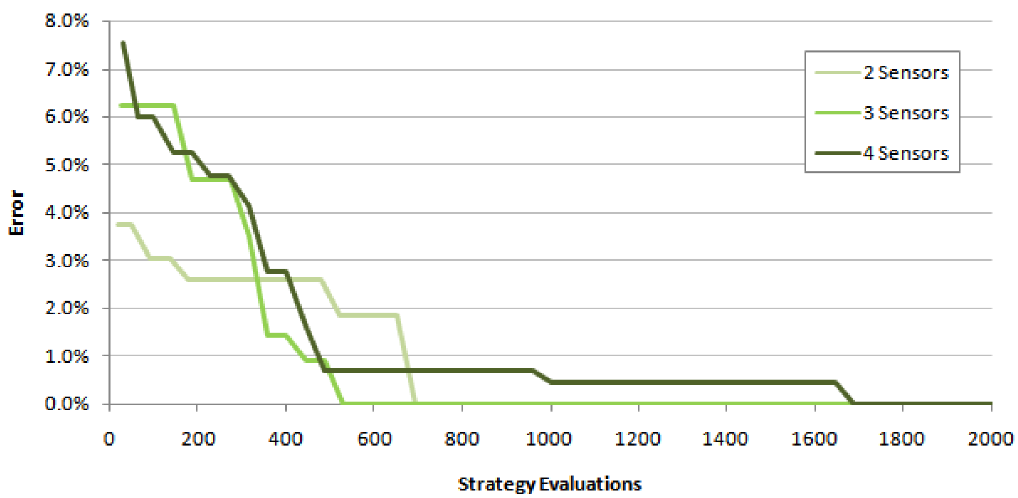

Figure 5.12: Algorithm convergence in Network II

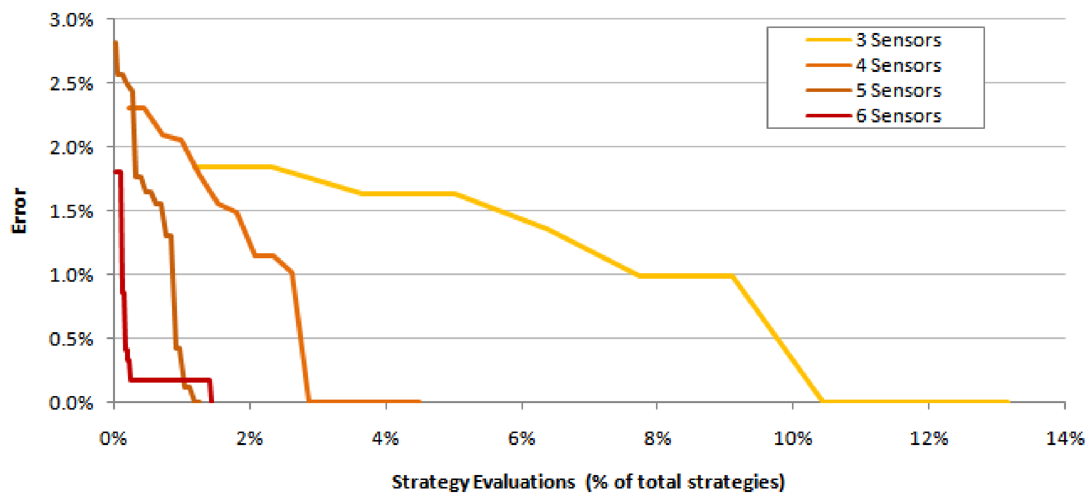

Figure 5.13: Algorithm convergence in Network I (Percent error)

equivalent to one or two cycles, after which the algorithm is terminated unless the optimal solution improves. This approach effectively led to the optimal solution in most analyzed cases.

The tables displaying the algorithm performance as a function of the percentage of total strategies evaluated suggest that in most cases the algorithm attains a solution within 1% of the optimal value after a number of evaluations equivalent to, at most, 10% of the possible strategies. The performance does not deteriorate with the network size, or in cases considering larger deployment strategies. However, as the number of required evaluations becomes larger, the effort involved in each of these computations becomes more relevant. The state partitioning algorithm presented in previous Sections alleviates the computational burden to a certain extent, but for very large sensor deployment strategies, heuristic procedures may be necessary in order to obtain solutions within a limited time frame.

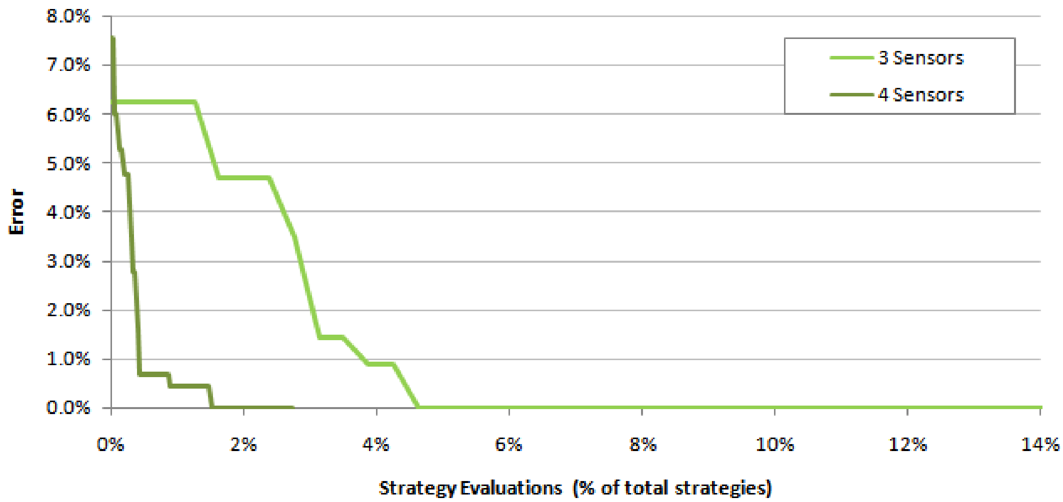

Figure 5.14: Algorithm convergence in Network II (Percent error)

5.2.3 Summary

This section introduced an adaptive memory Tabu search procedure for the solution of the optimal sensor deployment problem under information. The proposed algorithm considerable reduces the computational effort required to find near-optimal solutions, enabling the quick optimization of relatively large problems. The search procedure explores the solution space by moving from an existing solution to a new one within a predefined neighborhood. Moves are accomplished by swapping a specific number of elements between the existing solution and the neighborhood. The number of exchanged elements (move depth) varies cyclically throughout the search process, which results in alternating intensification and diversification phases that efficiently cover the solution space. The variation pattern is designed to take into account the problem characteristics, capturing the potential synergies behind the selection of specific combinations of links.

Identifying incoming links entails evaluating of all the candidate sets in the neighborhood. Given the potentially large size of such neighborhood, a randomized reduced neighborhood which includes a specific percentage of the potential candidates is utilized. The procedure implemented to generate such neighborhood accounts for a random component which avoids generating always the same subset of candidate solutions, but at the same time is systematic, preventing the duplication of candidate combinations. The algorithm terminates when a pre specified number of swap length cycles has been accomplished. Additional convergence criteria are utilized to refine the value of the solution near termination, increasing the chances of finding the exact optimal value.

The search process is guided using adaptive memory structures, or Tabu lists. These lists store recency and frequency information, reflecting how many times a particular link has been found to belong to a locally optimal solution in the short and long term. They are used to foster diversification, by avoiding the repeated re-incorporation of links which neighborhoods have already been explored.

Numerical tests conducted in two medium-size networks for several values of K, the number of deployed sensors, suggest a very satisfactory algorithmic performance. Optimal solutions were achieved by evaluating in average only 20% of the possible strategies in smaller cases (up to 10^3 candidate solutions), and only 3% of the candidate solutions for larger cases, with more than one hundred thousand candidate strategies. The results suggest that the algorithm is robust with respect to both, the network and the strategy size. However, when many sensors are considered, the methodology utilized to appraise each candidate solution plays a critical role, and heuristic approaches may be necessary to complement the state partitioning technique introduced in the previous section.

5.3 Implementation: Analyzing the impacts of sensor location on the performance of adaptive system-optimum routing strategies

In this section we implement the methodology described earlier to find the optimal deployment strategy of different numbers of sensors on two test networks, and discuss the results from various perspectives. In addition to the problems solved during the numerical tests conducted earlier, additional problems were solved utilizing the heuristic approach. These include the deployment of 5 and 6 sensors on Network II, and the deployment of 10 sensors on a modified version of network I with a smaller number of possible states per link (Table A.6).

The results displayed in Table 5.16 illustrate the beneficial impact of deploying an increasingly large number of sensors in Networks I and II. For these particular examples, the provision of information regarding the state of only 6 links leads to reductions in the system expected cost of 3% and 12%, respectively. The higher gains observed in Network II are likely to be a consequence of its large size and better connectivity, which translates into the availability of more alternative paths.

The practical value of the observed improvements clearly depends on what the link costs represent. Furthermore, the results corresponding to different networks and probability distributions are likely to vary widely in terms of the absolute gain, which motivates the qualitative type of analysis conducted in the remaining of this section. The proposed approach focuses on the properties of the solutions, which can be generalized and used to analyze the value of the novel models for future practical implementations.

Table 5.12 exhibits the marginal gain associated to the incorporation of each additional sensor, given by $\Delta G(i) = \frac{G(i)-G(i-1)}{G(i-1)}$. It is interesting to notice that this value does not exhibit a linearly decreasing trend. This illustrates problem properties already discussed in Section 4.4, which reflects the non-linear nature of the impacts of information. While the incorporation of a

	Network I			Network II		
K	$Z^{\mathcal{K}^*}$	$G(\%)$	$\Delta G(\%)$	$Z^{\mathcal{K}^*}$	$G(\%)$	$\Delta G(\%)$
0	152.5	-	-	64.5	-	-
1	151.79	0.47	0.47	61.77	4.23	4.23
2	150.22	1.49	1.03	61.06	5.33	1.09
3	148.94	2.34	0.84	60.52	6.17	0.84
4	148.27	2.77	0.44	59.79	7.30	1.13
5	148.02	2.93	0.16	59.40	7.91	0.61
6	147.66	3.18	0.24	56.83	11.89	3.98

Table 5.12: Impacts of information: Reductions in system expected cost

new sensor generally leads to an improvement in the system expected cost, the utilization of the provided information may be limited by the lack of measurements in complementary links. In other words, one may learn that link i exhibits a cost considerably lower than expected, but such information has no value if the costs on the remaining links on the path leading to i remain unknown. Eventually, the incorporation of enough additional sensors allows monitoring the network in such way that the potential benefits of the information collected at each individual link are fully realized. Once this is achieved, the marginal value of incorporating a new sensor is likely to stabilize, and may become zero.

Table 5.13 indicates the probabilities of optimally routed system assets facing expected costs above (O) and below (U) the expected cost under no information provision, ρ^0. Notice that the reported value is not the probability of actually paying a cost higher or lower than ρ^0, given that some link cost realizations remain unknown at the moment of making the corresponding routing decisions. However, the approximate values presented here provide some intuition regarding the characteristics of the solutions produced by the models. Even though in most of the cases the probabilities of facing expected costs lower than ρ^0 seems to increase with the number of deployed sensors, this needs not to be the case. For example, the value of U for the 6 sensors case in Network 2 is considerably lower than the corresponding value when

	Network I		Network II	
K	O	U	O	U
0	0.54	0.46	0.57	0.43
2	0.64	0.36	0.82	0.18
3	0.61	0.39	0.79	0.21
4	0.60	0.40	0.83	0.17
5	0.60	0.40	0.88	0.12
6	0.62	0.38	0.55	0.45

Table 5.13: Impacts of information: Probabilities of facing expected costs above and below the LEC

only 5 sensors are deployed. This reflects the fact that the reduction in system expected cost is a result of monitoring paths which have a positive probability of exhibiting a lower cost than expected. There is an inherent tradeoff between the magnitude of the possible gains, and the probability of attaining them. The model we propose is equally likely to select a path with high probabilities of leading to a moderate gain, than to choose a route which has insignificant chances of providing exceptional savings. Clearly, for some applications it may be relevant to avoid facing costs higher than a threshold value, and the objective function should be reformulated appropriately.

Figure 5.15displays the optimal location of, 1, 3 and 6 sensors on Network II. It also depicts all the paths which may be utilized during the corresponding asset routing under information. When one only sensor is available, the optimal solution places it on a link in $\mathcal{L}^0$, in such way that assets may be re-routed into the second best path if $\mathcal{L}^0$ exhibits a higher cost than expected. As more sensors become available, other alternative paths are measured. It is interesting to notice that when 6 sensors are deployed, none of them is placed along $\mathcal{L}^0$, and all the resources are devoted to identifying paths exhibiting lower costs than ρ^0.

The observed deployment and routing patterns suggest that information is first utilized to overcome the effects of high costs realizations on $\mathcal{L}^0$. However, if enough resources and alternative paths are available, sensors are used to unveil path cost realization which may be considerably lower than ρ^*. The

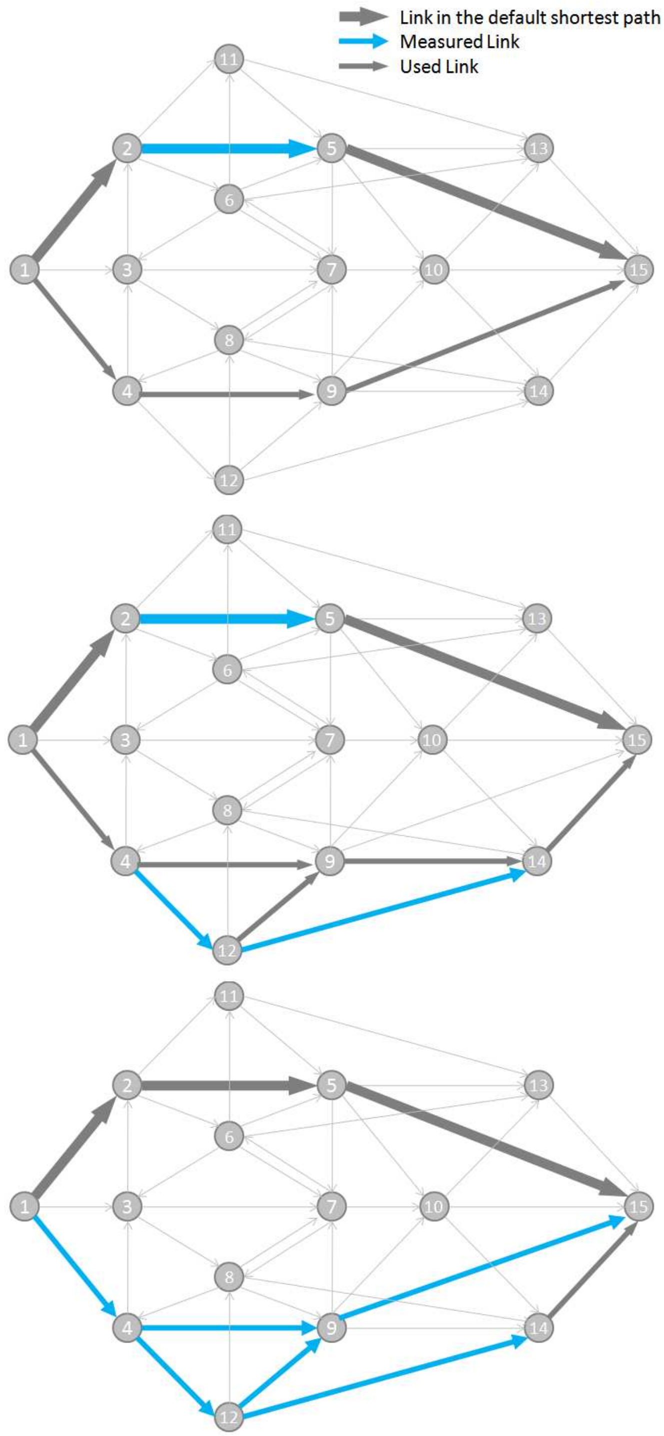

Figure 5.15: Optimal deployment of one, three and six sensors on Network II

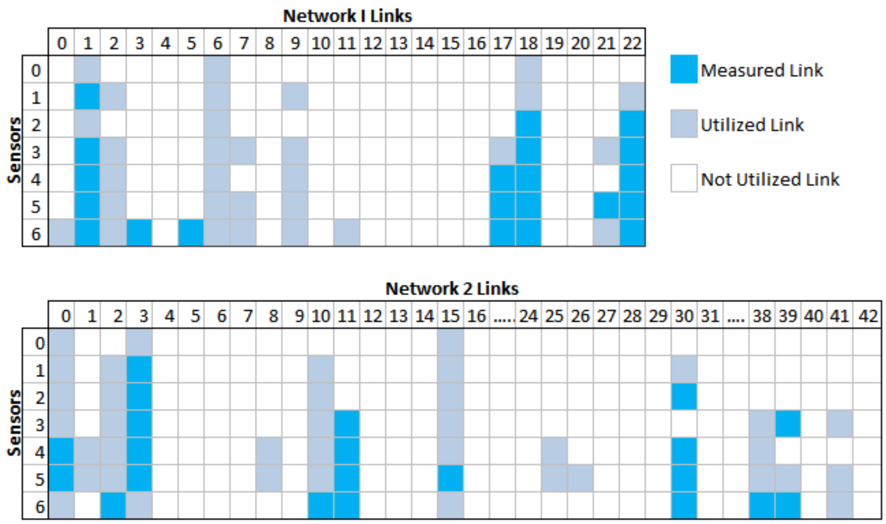

Figure 5.16: Optimal sensor deployment strategies and link utilization patterns

results also indicate that the deployment pattern depends on the cost variance (or standard deviation) on $\mathcal{L}^0$. On Network I, for which $\sigma = 31,5$ (compared to only 16.5 on Network 2), most of the strategies involve placing at least one sensor on $\mathcal{L}^0$. Figure 5.16 summarizes sensor deployment and path utilization patterns for Networks I and II, and strategy sizes ranging from 0 to 6.

Based on the former observations, two naive deployment strategies based on link variance were tested, and their results are presented in Table 5.14 and 5.15. The first of these strategies deploys the available sensors on the links with the highest variance. The resulting system expected, denoted $\rho^*(\sigma^2_{ALL})$, is considerably higher than the optimal values produced by our models (The loss is given by $L = \frac{\rho^*(\sigma^2_{ALL})-Z^*}{Z^*}$). The second strategy, which first deploys sensors on the highest variance links in $\mathcal{L}^0$, leads to solutions $\rho^*(\sigma^2_L)$ closer to the optimal values. This is most likely a consequence of measuring links in $\mathcal{L}^0$, which allows avoiding costs much higher than expected value if alternate paths with lower expected costs are available. The observed behavior suggested that "blind" sensor deployment strategies, which do not consider the impacts of the provided information, are less effective, and may have virtually no benefits for the system.

Network I

K	σ_{all}			$\sigma_{\mathcal{L}}$		
	ρ^*	$G\%$	$L\%$	ρ^*	$G\%$	$L\%$
0	152.5	0.00	-100	152.5	0.00	-100
1	152.5	0.00	-100	151.1	0.93	-37.4
2	151.8	0.45	-80.6	151.1	0.93	-60.0
3	150.7	1.18	-57.3	150.9	1.02	-63.0
4	150.7	1.19	-59.5	151.0	1.02	-65.1
5	150.7	1.19	-62.5	148.8	2.44	-23.3

Table 5.14: Results for maximum variance-based deployment strategies

Network II

K	σ_{all}			$\sigma_{\mathcal{L}}$		
	ρ^*	$G\%$	$L\%$	ρ^*	$G\%$	$L\%$
0	64.5	0.00	-100	64.50	0.00	-100
1	64.5	0.00	-100	61.39	4.82	-9.5
2	64.5	0.00	-100	61.39	4.82	-21.9
3	64.5	0.00	-100	61.39	4.82	-34.0
4	64.5	0.00	-100	61.39	4.82	-39.1
5	64.5	0.00	-100	60.89	5.68	-52.2

Table 5.15: Results for maximum variance-based deployment strategies

The results described in this section suggest that the models proposed in Chapter 4 can be used to improve the system performance and optimize the collection and utilization of information. Even though the absolute gains may vary widely depending on the characteristics of a particular network, the numerical analyses conducted for this application show the advantages of taking into account the utilization of information in the design of the corresponding data collection strategies. Naive approaches are suboptimal, and may result in resource investments which do not have any beneficial impact on the system. The analysis of optimal information collection patterns and the corresponding asset routing strategies also provides insights into the impacts of information in the utilization of a stochastic network. The models identify the network links which are critical in the connection of an origin-destination pair, which may be used as the basis to analyze the network performance under more complex behavioral assumptions. Furthermore, given that the availability of more information eventually leads to the utilization of a larger set of paths, the models can be used to study the design of information provision patterns fostering a more efficient network utilization.

5.4 Summary

This chapter presents, tests, and implements a methodology to find the sensor deployment pattern which optimizes the performance of adaptive system optimum routing strategies on a network with random arc costs. The models solved in this chapter identify sensor deployment strategies which account for the posterior utilization of the collected information. The corresponding decision variables are the links to be monitored, which define the set of perceived network states based on which adaptive routing decisions are made, and the routing strategies. In virtue of the integer nature of these variables, which represent a model constraint that cannot be relaxed (Section 4.2), the problem is combinatorial, and its solution requires the complete enumeration of all feasible sensor deployment strategies. The evaluation of each of these strategies, is also computationally challenging, involving in principle the computation of a shortest path for each perceived network state. The number of such states may be very large, depending on the available sensors and the characteristics of the link cost probability distributions.

The solution method is based on the fact that, given the assumptions presented in Section 4.2, the models may be solved by enumerating all feasible sensor/probe deployment strategies, and computing the corresponding expected costs under information. Such approach poses two main challenges: the large number of perceived states which need to be considered during the evaluation of a feasible deployment strategy, and the existence of a combinatorial number of strategies. The proposed solution technique deals with the first issue using state-partitioning principles, while the combinatorial problem is addressed heuristically, by implementing an adaptive memory Tabu search procedure.

The state-space partitioning algorithm, introduced in Section 5.1, is guided by rules developed specifically for the problems under study. These are used to reduce the number of shortest path computations required to find an optimal solution, mostly by appropriately selecting threshold values for the corresponding cost. Numerical experiments suggest that, in well connected networks, the algorithm may reduce the computational effort by up to 95%. The adaptive memory Tabu search procedure, presented in Section 5.2 explores the combinatorial solution space guided by short and long term memory structures. In the examples studied in Section 5.2.2 it found the optimal solution by evaluating between 3% and 20% of all the candidate solutions.

The performance of the proposed methodology is very satisfactory, and the results suggest that the effectiveness of the heuristic approach is not affected by the network size or the number of deployed sensors. However, the exact approach chosen to evaluate each feasible strategy may not be appropriate in cases involving many sensors, or in networks where the probability distribution functions exhibit a large number of states. A possible way of overcoming this problem may be the design of more complex partitioning rules, which may further reduce the number of shortest path computations per strategy evaluation. Additionally, shortest path re-optimization methods may be used to reduce the computational burden introduced by the large number of necessary evaluations. Appendix B provides a summary of such methodologies, which utilize the information provided by the solution of a shortest path problem to reduce the effort involved in re-solving the problem given some changes on the network costs.

The integrated methodology was implemented to the analysis of two medium sized networks, introduced in Alexopoulos [1997]. The results suggest that optimized sensor deployment strategies lead to improved adaptive system optimum decisions, leading to expected cost reductions ranging between 2% and 4%. The practical implication of the observed gains depends on the considered application. Furthermore, the magnitude of the results is likely to vary widely depending on the characteristics of the system under study. Nevertheless, the observed trends in the results are promising, and the proposed approach is up to 50% more efficient than methodologies which do not explicitly model the usage of information.

The analysis of the optimal information collection patterns and the corresponding asset routing strategies conducted in Section 5.3 provides interesting insights into the impacts of information on network utilization. The models identify the network links and paths which are critical in the connection of the considered origin-destination pair, which may be used as the basis to analyze the network performance under more complex behavioral assumptions. Furthermore, given that the availability of more information eventually leads to the utilization of a larger set of paths, the models can contribute to the study of information provision patterns allowing more efficient network utilization. The variations analyzed in this work involve a single origin destination pairs, a priori routing strategies, flow-independent link costs, and time invariant probability distributions. Further work in the area may relax these assumptions, in the search for a more flexible model which can be adapted to solve a variety of real world problems.

Chapter 6

Information Based System Optimum Assignment

The widespread adoption of wireless location technologies provides new means to automatically collect and distribute real time information from mobile assets, which measure the system state while they travel through the network. In transportation networks, traffic data collected from, and distributed to, moving vehicles has an enormous potential to improve the system performance by alleviating the negative impacts of uncertainty.

The availability of advanced technologies encourages the design of innovative approaches to traditional transportation problems, capable of exploiting the new sources of information. This chapter introduces a novel system-optimum network assignment paradigm which models the utilization of real time data to adjust system-optimum routing decisions, and takes advantage of the capability of assets to collect information as they travel through the network.

Vehicles traveling through a stochastic transportation network experience the realized state of each link they traverse, and therefore generate data about the network state. Thanks to the widespread adoption of Geographic Information Systems and other location-based technologies, the experienced cost data can be automatically collected, and eventually utilized for multiple purposes such as travel time prediction and network monitoring. In this context, the vehicles become probes, which sample the conditions throughout the network, with the potential to provide better coverage at a lower cost that traditional fixed traffic sensors (Cayford and Yim [2006.], W.L. et al. [2005]).

The Information-Based System Optimum (IBSO) assignment paradigm presented in this chapter shares the "cooperative" routing concept underlying traditional System Optimum (SO) assignment models, in virtue of which some assets may face higher costs than others in the search for an optimized system utilization. In the presence of uncertainty and information provision, one may consider that the impacts of an asset on the system cost are two-folded, including not only the cost the pay to traverse the network, but the information they collect along the way. As a result, some assets may be assigned to a higher-cost path than others in order to collect information which benefits the entire system. Such approach has a number of potential applications, including the cooperative deployment of emergency vehicles, military assets, and commercial vehicles. Furthermore, analyzing the properties and behavior of models incorporating the new paradigm may contribute to a better understanding of the impacts of information on the performance of transportation systems.

The approach proposed in this chapter is fundamentally new. The utilization of data provided by probe vehicles has been studied in the literature, and is discussed in Section 6.1. However, existing methodologies are centered on exogenously generated data, and they do not consider the possibility of selecting the routes along which information is collected. Furthermore, most of the existing models are not capable of measuring the effect of specific information collection strategies on the benefits derived from the corresponding data.

The mathematical model proposed in this chapter (Sections 6.2 and 6.3) captures the impact of different information collection patterns on the performance of adaptive system optimum routing strategies. By allowing the utilization of some of the system assets as probes, it implicitly captures the trade-offs between the cost and value of information, which is of the utmost interest for practical purposes. It also presents other interesting properties, which are discussed in Section 6.4. The model is implemented to the analyses of several example problems (Section 6.6), and the result analyses suggests that the proposed approach may take advantage of new data sources to improve the system performance. The methodology used to solve the problem is an adaptation of the techniques presented in Chapter 5. The numerical experiments conducted using this approach also illustrates the problem behavior and properties, providing insightful information for the future design of practical application models and more efficient solution procedures.

6.1 Literature Review: Using probe vehicle data in transportation networks

Vehicles traveling through a stochastic transportation network experience the realized state of each link they traverse, and therefore collect information about the network state. Thanks to the widespread adoption of Geographic Information Systems and other location-based technologies, the experienced cost data can be automatically collected, and eventually utilized for multiple purposes such as travel time prediction and network monitoring. In this context, the vehicles become probes, which sample the conditions throughout the network, with the potential to provide better coverage at a lower cost that traditional fixed traffic sensors (Cayford and Yim [2006.], W.L. et al. [2005]). Kim and Ra.Cayford [2000] study the utilization of cell phone and GPS data for traffic monitoring, and concluding that systems providing a location accuracy of 20 meters or less are adequate for such purpose. The data available in the San Francisco Bay Area at the time this effort was conducted was enough to covered 99% of the major freeways and arterials.

A number of studies have been conducted in order to analyze the potential utilization of wireless location information for system monitoring purposes. Most of the methodologies consider that probe vehicles provide a representative sample of the traffic conditions experienced on the network, and apply statistical techniques to compute the average value of the condition of interest, typically speed. Performance is generally assessed in terms of system coverage and estimation accuracy (Fontaine and Smith [2005]), and model parameters include sample size and frequency of the sampling. W.L. et al. [2005] study the sample size necessary to produce good estimates of travel time and congestion accounting for the practical limitations imposed by wireless networks and various transportation network parameters. They indicate the need to adjust sample size and frequency based on traffic conditions and vehicle characteristics.

Fontaine and Smith [2005] analyze the effect of network characteristics and sampling methodologies on the quality of the speed estimations generated using wireless location technologies. They implement a simulation-based approach, and their findings highlight the importance of the map-matching procedure used to identify the actual location of a vehicle in the network. The authors also remark the need for estimation models able to adapt to the network conditions by adapting the sample size, as well as the temporal and spatial characteristics of the sampling procedure. Kwon et al. [2007] compare the performance of probe-based data and loop detectors information in the generation of congestion information, and develop a methodology which produces reliable estimations of the conditions in urban freeways when 4 to 6 days of good probe data is available. Their findings suggest that this is as efficient as estimating congestion based on loop detectors spaced half a mile.

Commercial and public transportation vehicles are typically equipped with GIS devices (Automatic Vehicle Locators-AVL), and several authors (e.g. Chakroborty and Kikuchi [2004], Cathey and Dailey [2002], Dailey and Cathey [2006], Tantiyanugulchai and Bertini [2003]) analyze the utilization of the information they provide for travel time prediction. In this approach, a fundamental issue is the relationship between the probe speed and the average system speed. Chakroborty and Kikuchi [2004] conduct studies comparing bus travel time to the travel time experienced by passenger cars. The corresponding findings suggest that the difference between the two magnitudes is relatively stable, and therefore mathematical expressions can be developed to derive the average speed of passenger cars based on bus probe data. Tantiyanugulchai and Bertini [2003] study the same problem in Portland, Oregon, finding that the average speed of a regular vehicle is 1.3 that of the probe bus speed. Dailey and Cathey [2006] analyzes the utilization of buses in the Seattle area to monitor traffic conditions and improve traffic management strategies, developing a methodology which estimates congestion and travel speeds with an accuracy comparable to that obtained from static sensors.

Moore II et al. [2001] study the utilization of patrol cars in California to produce travel time estimates. Their findings suggest that, in general, the speed of patrol cars is not a good approximation of the prevalent speeds on the freeway sections they traverse, and that the covariance of both magnitudes is somewhat erratic, complicating the utilization of the corresponding data.

An upcoming approach to employ passenger cars as probes is based on cell phone location data (Cayford and Johnson [2003], Cayford and Yim [2006.], Bar-Gera [2007], Foo et al. [2006], Jin et al. [2007]). Cayford and Yim [2006.] makes use of the technology developed to track emergency cell phone calls in order to monitor the transportation network state in Tampa, Florida, generating average speed data for 98% of all the major freeways in the area under study with relatively low estimation errors (5-10 mph). Bar-Gera [2007] describes a similar application in Israel, concluding that the predictions based on cell phone data had a comparable accuracy to those based on dual loop detector information. From a different perspective, Davies et al. [2006] explore the utilization of GIS data from passenger for updating and correcting road maps, finding that the methodology is promising and provides reasonably accurate results for roads with relatively dense GPS readings.

The review conducted in this section suggests that there are multiple sources of online transportation data, and work is being conducted towards the efficient utilization of the corresponding information for system monitoring purposes. However, there's virtually no research analyzing the utilization of probe-vehicle data for route guidance purposes. Furthermore, the effects of the specific routes followed by the probes on the quality and usefulness of the corresponding information have not been explored. The later is of great interest from two perspectives: firstly, the actual route followed by some of the probes, such as buses and commercial vehicles, may be optimized in order to generate the largest possible system benefits. Additionally, understanding the impacts of different information collection patterns allows identifying critical data sources, eventually reducing the amount of information that needs to be processed for specific purposes.

6.2 The Information-Based System Optimum assignment paradigm

This section conceptually describes the Information Based System Optimum (IBSO) paradigm introduced in this chapter, which shares some common elements with the traditional system-optimum assignment (SO) problem. The later finds the optimal flow patterns satisfying given origin-destination (OD) demands on a network where link costs are a convex function of the corresponding flow. Optimality is defined as the minimization of the total system cost, which is equal to the summation of the cost paid by each individual vehicle. Sheffi [1985] provides a rigorous mathematical formulation of this problem. One of the most notable characteristics of an SO assignment strategy is that the selection of optimal routes takes into account both, the cost faced by an asset when utilizing a path, and the impact that its presence on the path the has on the cost paid by the remaining assets which utilize it . This property, reflected in the problem's optimality conditions Sheffi [1985], is the one that guarantees that an optimized system-level performance is achieved, at the cost of allowing some assets to face higher costs than others. Moreover, the path flow patterns in an optimal SO solution are such that it would be possible for some vehicles to switch to a different path and incur in a lesser cost. However, that "selfish" behavior would have a negative impact on the assets already assigned into the alternative path, and worsen the system performance.

The Information Based System Optimum (IBSO) assignment shares SO's "unselfish" routing perspective. In the presence of uncertainty and information provision, one may consider that the impacts of an asset on the system cost are two-folded, including not only the cost the pay to traverse the network, but the information they collect along the way. As a result, some assets may be assigned to a higher-cost path than others in order to collect information which benefits the entire system.

Within a stochastic context, the cost on links and paths is expressed in terms of expectations. In the absence of additional information (and assuming that link costs are flow-independent), the shortest expected cost path is a reasonable routing alternative which minimizes the system expected cost. However, such path is not necessary the least expensive under every possible network realization and the system may benefit from learning the actual cost on one or more paths. The basic concept underlying the IBSO assignment paradigm is that a subset of system assets may be regarded as probes which can measure and communicate path cost realizations. Probes may be assigned to routes exhibiting a higher expected cost, in an attempt to find lower cost realizations than may benefit the entire system. However, the assets utilized as probes are part of the system, and therefore the additional cost paid to collect information should be compensated by the benefits experienced by the remaining assets. The change in system-expected cost introduced by the utilization of an additional asset as a probe reflects the trade-offs between the value of information and the cost paid to acquire it.

From a modeling perspective, the collection of information is modeled as a cost change on the links traversed by probes. The benefits of information are accrued by implementing an adaptive assignment scheme, which allows modifying optimal routing strategies based on the cost realizations measured by the probes.

The concept of IBSO assignment may lead to the formulation of a multitude of problems, depending on the assumptions regarding information collection and utilization patterns, and the characteristics of the considered system (Section 2). The instance analyzed in this chapter adopts a Serial and

Sequential (SS) probe deployment approach, which implies that all the assets utilized as probes enter the system together, and the routing decision for the regular assets is not made until the corresponding information is retrieved. Notice that minimizing the system expected cost is not the only desirable objective function. For applications such that cost reliability is highly valued, formulations minimizing the variability of the experienced cost with respect to a fixed target, or incorporating a maximum admissible cost under any scenario may be more appropriate, and will be the subject of further research.

6.3 Problem formulation

This section discusses the mathematical formulation of the Information-Based System-Optimum (IBSO) assignment problem under a Serial and Sequential (SS) probe deployment strategy. This problem involves finding the optimal routes to be followed by the system assets assuming that a subset of these are utilized as probes, which enter the network first and monitor the conditions on the links they traverse. Non-probe assets, also referred to as regular assets, are optimally routed based on the information retrieved by the probes once the latter reach their destination. The process is two-tiered, and all the probes are deployed simultaneously into the network, in virtue of which they can not take advantage of the information collected by their peers.

The problem formulated in this section lends itself to be modeled as a bi-level stochastic program which is able to capture the underlying sequential decision making process. The proposed model follows a path-based approach, in virtue of which the objective function and corresponding marginal costs are easier to interpret and analyze. The later is fundamental in order to develop a better understanding of the problem properties and behavior. The notation utilized in this chapter is slightly different to the one introduced in Chapter 5, and it is described below. Let $G(\mathbf{N}, \mathbf{M})$ represent a network with a set of N nodes $i \in \mathbf{N}$ and the corresponding set of M arcs $ij \in \mathbf{M}$, characterized by an infinite capacity and random weights $\tilde{c}_{ij}$. Assume that the latter are independent of the corresponding link flows, and that they follow a discrete probability distribution consisting of a finite number of states $s_{ij} \in S_{ij}$, with probability of occurrence $p_{s_{ij}} s.t.$ $\sum_{s \in S_{ij}} p_s = 1 \ \forall \ ij \in A$. For notational simplicity, the subscript in s_{ij}will be suppressed whenever it can be inferred from the context. Additionally, a single index may be used to denote a link when its origin and destination are not relevant.

Let $r \geq |S_{kl}| \ \forall \ kl \in A$ represent the maximum number of states observed across all links. Define c_{ij}^s as the cost realization corresponding to state $s \in S_{ij}$, and denote $\mu_{ij} = \sum_s p_s \cdot c_{ij}^s$ the expected cost of a link $ij \in A$. Network states are a result of the corresponding link states, and are represented using $m-$ dimensional vectors, $\mathbf{w} \in \mathbf{W}$. Let $s_{ij}^{\mathbf{w}}$ be the state on link ij under state $\mathbf{w}$, and $c_{ij}^{\mathbf{w}} = c_{ij}^{s_{ij}^{\mathbf{w}}}$the corresponding link cost. Under the assumption of independent and uncorrelated link cost functions, the probability of a network state can be computed as $p_w = \prod_{ij \in A} p_{s_{ij}^{\mathbf{w}}}$. Notice that under the previous assumption, the cardinality of $\mathbf{W}$ is$|\mathbf{W}| = \prod_{ij \in A} |S_{ij}|$, and it grows exponentially with M.

Denote K the total number of assets to be utilized as probes, and $\mathbf{P}_{s-t}$ the set of all paths P^h connecting origin-destination pair $s-t$, each of which can be considered as a subset of $\mathbf{M}$ ($P^h \subseteq A$). The cardinality of $\mathbf{P}$ (the sub index will be omitted given the assumption of a unique OD pair) depends on the network topology, and grows as an exponential function of M in a complete network. Let $\tilde{\kappa}_{P^h} = \sum_{ij \in P^{hi}} \tilde{c}_{ij}$ be the cost of a path, computed as the summation of the costs of its links, and denote $\kappa_{P^h}^{\mathbf{w}} = \sum_{ij \in P} c_{ij}^{\mathbf{w}}$ the path cost realization corresponding to network state $\mathbf{w}$. The expected cost of a path can be computed as $\psi_{P^j} = \sum_{ij \in h} \mu_{ij}$. Binary decision variables f_{P^h} and g_{P^h} are used to represent the utilization of path P^h by probe vehicles and regular assets, respectively. There is a direct correspondence between these variables and the link flow variables introduced in Chapter 5, given by equations 6.1 and 6.2, where δ_{ij}^h are link-path incidence parameters, equal to one if link ij is in path P^h and to zero otherwise.

$$x_{ij} = \sum_{P^h \in \mathbf{H}} f_{P^h} \cdot \delta_{ij}^{P^h} \; \forall \, ij \in A \tag{6.1}$$

$$y_{ij} = \sum_{P^h \in \mathbf{H}} g_{P^h} \cdot \delta_{ij}^{P^h} \; \forall \, ij \in A \tag{6.2}$$

Equations 6.3 to 6.9 present the bi-level formulation for a problem instance utilizing one asset as probe in a system with T regular assets.

$$min\, E[z(f, \tilde{\mathbf{c}})] \tag{6.3}$$

$$\sum_{P^h \in \mathbf{H}} f_{P^h} = 1 \tag{6.4}$$

$$f_{P^h} \in \{0, 1\} \tag{6.5}$$

$$z(x, \tilde{\mathbf{c}}) = min \sum_{P^h \in \mathbf{H}} f_{P^h} \cdot \tilde{\kappa}_{P^h} \tag{6.6}$$

$$+ \sum_{P^h \in \mathbf{H}} f_{P^h} \cdot T \cdot g_{P^h} \cdot \left(\sum_{ij \in A} \delta_{ij}^{P^h} \cdot (\tilde{c}_{ij} \cdot \sum_{P^i \in H} \delta_{ij}^{P^i} \cdot f_{P^i} + \mu_{ij} \cdot (1 - \sum_{P^i \in H} \delta_{ij}^{P^i} \cdot f_{P^i})) \right) \tag{6.7}$$

$$\sum_{P^h \in \mathbf{H}} g_{P^h} = 1 \tag{6.8}$$

$$g_{P^h} \in \{0, 1\} \tag{6.9}$$

The second-stage problem is solved for each possible network state $\mathbf{w} \in \mathbf{W}$ and, under the assumption of uncapacitated links and flow-independent link costs, it can be reduced to a shortest path problem. For fixed values of $f_{P^h}$the first term of the lower-level objective function is constant, and the same applies to the expression $\tilde{c}_{ij} \cdot \sum_{P^i \in H} \delta_{ij}^{P^i} \cdot f_{P^i} + \mu_{ij} \cdot (1 - \sum_{P^i \in H} \delta_{ij}^{P^i} \cdot f_{P^i})$, which defines the **cost of link conditional** on the available information $\widetilde{c}_{ij}|_{\mathcal{I}}$. The latter represents the cost of a link given the information $\mathcal{I}$ obtained by the probe asset. The conditional cost of link ij is equal to the corresponding expected cost, unless ij belongs to any of the paths monitored by the probe asset. The effect of information is therefore modeled by replacing the estimator of an uncertain arc cost (μ_{ij}) by the cost realization learnt by the probe vehicles. The conditional cost of a path is defined by $\kappa_{P^h}|_{\mathcal{I}} = \sum_{ij \in A} \delta_{ij}^{P^h} \cdot \widetilde{c}_{ij}|_{\mathcal{I}}$.

The previous formulation may be reduced to a single level problem by expanding the expected cost expression. The new objective function is provided in equation 6.10, and introduces and additional index on the flow variables corresponding to the non-equipped assets, in order to keep track of network state to which they correspond. The one stage problem is subject to equations 6.4, 6.5 ,6.9, and $\sum_{P^h \in \mathbf{H}} g_{P^h}^{\mathbf{w}} = 1$, $g_{P^h}^{\mathbf{w}} \in \{0, 1, \} \, \forall \, P^h \in \mathbf{H}$, $\mathbf{w} \in \mathbf{W}$.

$$min \sum_{\mathbf{w} \in \mathbf{W}} p^{\mathbf{w}} \cdot \sum_{P^h \in \mathbf{H}} \{ f_{P^h} \cdot \kappa_{P^h}^{\mathbf{w}} + T \cdot g_{P^h}^{\mathbf{w}} \cdot \kappa_{P^h}^{\mathbf{w}} | \mathcal{I} \} \tag{6.10}$$

The objective function presented in equation 6.10 may be transformed by algebraic manipulations into Equation 6.11, which provides a more compact expression of the problem. In such equation, φ_{P^h} is the expected cost paid by the probe, and $\lambda_{P^h} = \sum_{\mathbf{w} \in \mathbf{W}} p^{\mathbf{w}} \cdot \sum_{P^h \in \mathbf{H}} T \cdot g_{P^h}^{\mathbf{w}} \cdot \kappa_{P^h}^{\mathbf{w}} | \mathcal{I}$ is the expected cost faced by the regular assets when information is collected on path P^h.

$$Z^{\mathcal{K}} = \varphi_{P_h} + T \cdot \lambda_{P^h} \tag{6.11}$$

Notice that this formulation is valid under the assumption that $K = 1$, in virtue of which $\sum_{P^h \in \mathbf{H}} \delta_{ij}^{P^h} \in \{0, 1\}$. In order to handle the general case $K \geq 1$, a valid alternative is to to replace $\sum_h \delta_{ij}^h \cdot f_h$ with variables d_{ij} such that $d_{ij} \leq \sum_h \delta_{ij}^h \cdot f_h \, \forall \, ij \in A$, $d_{ij} > f_h \cdot \delta_{ij}^h \, \forall \, h \in H$, and $d_{ij} \in \{0, 1\}$.

Another option is to create an aggregate decision variable $\mathcal{K}^j(K)$ which represents a combination of K paths in $\mathbf{P}$. Such combinations represent a feasible deployment strategies for the assets utilized as probes, and belong to the set $\mathbf{K}$ which contains all the possible combinations of K elements out of $|\mathbf{P}|$. Let $v_{\mathcal{K}^j}$ be a new first level decision variable, representing the assignment of assets used as probes to the paths combined by $\mathcal{K}^j$. Also, define the link-strategy incidence parameter ϕ_{ij}^q to be equal to one if link ij belongs to any of the paths included in $\mathcal{K}^j$, and to zero otherwise. The formulation obtained replacing f_{P^h} and $\delta_{ij}^{P^h}$ in equation 6.6 by $v_{\mathcal{K}^j}$ and ϕ_{ij}^{q} , respectively, is valid for any value of K. Equation 6.11 may be reformulated as $Z^{\mathcal{K}} = \sum_{P^h \in \mathcal{K}_j} \varphi^{P^h} + T \cdot \lambda_{\mathcal{K}_j}$, where the first term represents the costs paid by all probes, and the second term is the expected cost faced by the regular assets given all available information.

The former approach clearly involves a very large number of variables, proportional not only to the number of states, but to the number of paths and strategies.

Similarly to what was observed in 4.2, the proposed mathematical formulations are insightful, but unlikely to be used directly for the solution procedure.

6.3.1 The marginal value of information

Using finite differences, one may compute the marginal impact on the system cost of utilizing a new asset as a probe assigned to path P^1. The corresponding value may be considered as the marginal benefit of collecting information from path P^1, and it is given by equation 6.14, derived from Equation 6.10 as presented in Equations 6.12 and 6.13. Given that $g_{P^h}^{w}$ is a function of f_{P^h}, equation 6.10 may be regarded as the product of two functions of the same variable, and the product rule is applied.

$$\frac{\Delta z(f_{P^j}, g_{P^j}^{\mathbf{w}}(f_{P^j})}{\Delta f_{P^1}} = \frac{\Delta z(f_{P^h}, g_{P^j}^{\mathbf{w}}(f_{P^h})}{\Delta f_{P^1}} \cdot g_{P^j}^{w}(f_{P^j}) + \sum_{P^j \in \mathbf{P}} \frac{\Delta z(f_{P^j}, g_{P^j}^{\mathbf{w}}(f_{P^h})}{\Delta g_{P^j}^{\mathbf{w}}} \cdot z \cdot \frac{\Delta g_{P^j}^{\mathbf{w}}(f_{P^j})}{\Delta f_{P^1}} \tag{6.12}$$

$$\frac{\Delta z(f_{P^h}, g^w_{P^h}(f_{P^h})}{\Delta f_{P^1}} = \sum_{\mathbf{w}\in\mathbf{W}} p^{\mathbf{w}} \cdot \kappa^{\mathbf{w}}_{P^1} + T \cdot \sum_{\mathbf{w}\in\mathbf{W}} p^{\mathbf{w}} \cdot \sum_{P^i\in\mathbf{P}} \left(\frac{\Delta g^{\mathbf{w}}_{P^j}}{\Delta f_{P^1}} \cdot \sum_{ij\in\mathbf{M}} \delta^{P^j}_{ij} \cdot (c^{\mathbf{w}}_{ij} \cdot \sum_{P^k\in\mathbf{P}} \delta^{P^k}_{ij} f^0_{P^k}) \right)$$
$$+T \cdot \sum_{\mathbf{w}\in\mathbf{W}} p^{\mathbf{w}} \cdot \sum_{P^i\in\mathbf{P}} \left(g^{\mathbf{w}}_{P^j} \cdot \sum_{ij\in\mathbf{M}} \delta^{P^j}_{ij} \cdot c^{\mathbf{w}}_{ij} \cdot \delta^{P^1}_{ij} \right) + T \cdot \sum_{\mathbf{w}\in\mathbf{W}} p^{\mathbf{w}} \cdot \sum_{P^i\in\mathbf{P}} \left(\frac{\Delta g^{\mathbf{w}}_{P^j}}{\Delta f_{P^1}} \cdot \sum_{ij\in\mathbf{M}} \delta^{P^j}_{ij} \cdot \mu_{ij} \right)$$
$$-T \cdot \sum_{\mathbf{w}\in\mathbf{W}} p^{\mathbf{w}} \cdot \sum_{P^i\in\mathbf{P}} \left(\frac{\Delta g^{\mathbf{w}}_{P^j}}{\Delta f_{P^1}} \cdot \sum_{ij\in\mathbf{M}} \left(\delta^{P^j}_{ij} \cdot \sum_{P^k\in\mathbf{P}} \delta^{P^k}_{ij} f^0_{P^k} \right) \right)$$
$$- \sum_{\mathbf{w}\in\mathbf{W}} p^{\mathbf{w}} \cdot \sum_{P^i\in\mathbf{P}} \left(g^{\mathbf{w}}_{P^j} \cdot \sum_{ij\in\mathbf{M}} \delta^{P^j}_{ij} \cdot \mu_{ij} \cdot \delta^{P^1}_{ij} \right) \tag{6.13}$$

$$\sum_{\mathbf{w}\in\mathbf{W}} p^{\mathbf{w}} \cdot \kappa^{\mathbf{w}}_{P^1} + \sum_{\mathbf{w}\in\mathbf{W}} p^{\mathbf{w}} \cdot \sum_{ij\in\mathbf{M}} \delta^{P^1}_{ij} \cdot \left(\mu_{ij} + T \cdot \sum_{P^i\in\mathbf{P}} g^{\mathbf{w}}_{P^j} \cdot \delta^{P^j}_{ij} \cdot (c^{\mathbf{w}}_{ij} - \mu_{ij}) \right) +$$
$$- \sum_{\mathbf{w}\in\mathbf{W}} p^{\mathbf{w}} \cdot \sum_{P^i\in\mathbf{P}} \left(\frac{\Delta g^{\mathbf{w}}_{P^j}}{\Delta f_{P^1}} \cdot \sum_{ij\in\mathbf{M}} \delta^{P^j}_{ij} \left(c^{\mathbf{w}}_{ij} \cdot \sum_{P^k\in\mathbf{P}} \delta^{P^k}_{ij} f^0_{P^k} - \mu_{ij}(1 - \sum_{P^k\in\mathbf{P}} \delta^{P^k}_{ij} f^0_{P^k}) \right) \right) \tag{6.14}$$

In Equation 6.14, $\frac{\Delta g^{\mathbf{w}}_{Pj}}{\Delta f_{P1}}$ represents the adjustments to the routing strategy of non-probe assets performed based on the availability of information from path P^1. It is equal to 0 for all values of k when monitoring of P^1 does not lead to any change in the routing decision under $\mathbf{w}$. Otherwise, it is equal to 1 for the new optimal path $\mathcal{L}^{\mathbf{w}}$, and to -1 for $\mathcal{L}^{\mathbf{w}-}$, the path which was optimal before collecting information from path P^1. Equation 6.14 may be simplified by considering that $g^{\mathbf{w}}_{Pj} = 1$ only for the optimal path under $\mathbf{w}$ ($\mathcal{L}^{\mathbf{w}}$). Also, in order to ease the interpretation of the marginal cost expression, assume that a single asset, which was previously not part of the system, is utilized as a probe. The later implies that that $f^0_{P^k} = 0 \ \forall \ k \in \mathbf{P}$. For each state $\mathbf{w}$, the corresponding term in Equation 6.14 may be reduced to the expression displayed in Equation 6.15, where $\Lambda = 1$ if $\frac{\Delta g^{\mathbf{w}}_{\mathcal{L}^{\mathbf{w}}}}{\Delta f_{P^1}} = 1$, and $\Lambda = 0$ otherwise.

$$\sum_{ij \in \mathbf{M}} \delta_{ij}^{P^1} \cdot \left(c_{ij}^{\mathbf{w}} + \delta_{ij}^{\mathcal{L}^{\mathbf{w}}} \cdot T \cdot \left((c_{ij}^{\mathbf{w}} - \mu_{ij}) + \frac{\Delta g_{\mathcal{L}^{\mathbf{w}}}^{\mathbf{w}}}{\Delta f_{P^1}} \cdot \mu_{ij} \right) - \delta_{ij}^{\mathcal{L}^{\mathbf{w}-}} \cdot T \cdot \mu_{ij} \right)$$
$$+ \sum_{ij \in \mathbf{M}, ij \notin P^1} \Lambda \cdot \left(\left(\delta_{ij}^{\mathcal{L}^{\mathbf{w}}} - \delta_{ij}^{\mathcal{L}^{\mathbf{w}-}} \right) \cdot \mu_{ij} \right) \quad (6.15)$$

The first summation in Equation 6.15 represents the change in the system cost directly related to the utilization of path P^1 (local-level information impacts). The second term measures the network-level impact of the information collected along P^1, which is a result of the adjustments introduced into the routing decisions concerning regular assets.

The local-level marginal costs are determined by large by the cost experienced on links which belong to $\mathcal{L}^{\mathbf{w}} \cap P^1$, which are such that $\delta_{ij}^{\mathcal{L}^{\mathbf{w}}} \cdot \delta_{ij}^{P^1} = 1$. If none of the links on P^1 belongs to $\mathcal{L}^{\mathbf{w}}$, the local-level marginal impact is simply the cost faced by the probe.

When $\mathcal{L}^{\mathbf{w}} \cap P^1 \neq \emptyset$, the local-level marginal impact may take three types of values on each link in P^1:

- Measuring Impact $\left(c_{ij}^{\mathbf{w}} + T \cdot (c_{ij}^{\mathbf{w}} - \mu_{ij})\right)$: This value is achieved when link ij was part of the optimal solution under $\mathbf{w}$ before deploying a probe on P^1, and it remains on the shortest path given the new information $\left(\frac{\Delta g_{\mathcal{L}^{\mathbf{w}}}^{\mathbf{w}}}{\Delta f_{P^1}} = 0,\ \delta_{ij}^{\mathcal{L}^{\mathbf{w}}} = 0,\ \delta_{ij}^{\mathcal{L}^{\mathbf{w}-}} = 0\right)$. It reflects the difference between the expected cost and the observed realization at the corresponding link, which is a gain (or loss) experienced by all the regular assets in the system.

- Incorporation Impact $\left(c_{ij}^{\mathbf{w}} + T \cdot c_{ij}^{\mathbf{w}}\right)$: This impact reflects the contribution to the marginal cost of links which were not part of the optimal route under state $\mathbf{w}$ before P^1 was monitored $\left(\frac{\Delta g_{\mathcal{L}^{\mathbf{w}}}^{\mathbf{w}}}{\Delta f_{P^1}} = 1,\ \delta_{ij}^{\mathcal{L}^{\mathbf{w}}} = 1,\ \delta_{ij}^{\mathcal{L}^{\mathbf{w}-}} = 0\right)$. It reflects the cost faced by all the assets utilizing the link.

- Removal Impact $(c_{ij} - T \cdot \mu_{ij})$: This impact is a measure of the change in the system cost resulting from removing links in $E \cap P^1$ from the optimal solution under $\mathbf{w}$ given the information collected on P^1 $\left(\frac{\Delta g^{\mathbf{w}}_{\mathcal{L}^{\mathbf{w}}}}{\Delta f_{P1}} = 1,\ \delta_{ij}^{\mathcal{L}^{\mathbf{w}}} = 0,\ \delta_{ij}^{\mathcal{L}^{\mathbf{w}-}} = 1\right)$. It is actually a reflection of the change in the routing strategy brought about by the newly available information.

Notice that Equation 6.15 may have a positive value, and therefore the utilization of an additional asset as a probe may increase the system expected cost. An example of this a a case such that P^1 is not utilized under any perceived state $\mathbf{w}$, and the corresponding information does not lead to changes in the routing strategies of regular assets $\left(\frac{\Delta g^{w}_{\mathcal{L}^{\mathbf{w}}}}{\Delta f_{P1}} = 0\ \forall \mathbf{w} \in \mathbf{W}\right)$, which leads to a marginal cost equal to $\varphi = \sum_{ij \in P1} \mu_{ij}$, the expected cost faced by the probe. Similar results are obtained if we derive the marginal costs assuming that the asset utilized as a probe was formerly a regular asset. Under such case, the second term is multiplied by $T-1$, and a third term, equal to $-\rho^0$ is added to Equation 6.15. Under such setting, the marginal cost may still be positive, with a value of $\varphi - \rho^0 = \Delta\varphi$, which represents the additional cost faced by the probe with respect to being routed on the default shortest path.

The marginal cost formulations presented in this section are useful to understand how information affects the system performance, and contribute to a better understanding of the problem properties. Notice that the definitions introduced above suggest that, in order for information to be valuable for the system, it must leads to changes into the routing strategies of regular assets.

6.4 Problem properties

The problem properties present some similarities to those observed in Section 4.4, given that the utilization of assets as probes ultimately translates into the availability of information form a subset of links. In virtue of this, the first and third properties enunciated in Section 4.4 are valid for the subset of links $ij \in \mathcal{K}^*$. The following properties are specific to the collection of information along paths.

- **Property 1:** The marginal benefit obtained from utilizing an additional asset as a probe routed on path P^hmay be negative. This was proved in Section 6.3.1.

- **Definition 1:** A strategy $\mathcal{K}_j$ consisting of K paths $i \in \mathcal{K}_j$ is considered to be **efficient** if there exists a finite value of $T = T_v^{\mathcal{K}_j}$ such that $Z^{\mathcal{K}_j} = \sum_{i\in\mathcal{K}_j} \varphi_i + \lambda_{\mathcal{K}_j} \cdot T_v^{\mathcal{K}_j} \leq S$, where $S = \rho^0 \cdot (T + K)$ is the system expected cost under a traditional SO assignment given the problem assumptions. If $\lambda_j < \rho^0$, $T_v^{\mathcal{K}_j}$ has a finite value given by equation 6.16. Such value represents the minimum number of non-probe assets which justifies the additional expected cost faced by probes when these are assigned to the paths in $\mathcal{K}_i$.

$$T_v^{\mathcal{K}_j} \geq \frac{K \cdot (\rho^0 - \sum_{i\in\mathcal{K}_j} \varphi_i)}{(\rho^0 - \lambda_{\mathcal{K}_j})} \tag{6.16}$$

- **Property 2:** Let $\mathbf{K}^K$ be the set of all efficient strategies of size K. The optimal strategy when $T \to \infty$ is $\mathcal{K}_E \in \mathbf{K}^K :\ \lambda_{\mathcal{K}_E} \leq \lambda_{\mathcal{K}_j}\ \forall\ \mathcal{K}_j \in \mathbf{K}^K$, the strategy exhibiting the lowest value of $\lambda_{\mathcal{K}_j}$.

Proof: We show that for every pair-wise comparison of efficient strategies the one with the lowest value of $\lambda_{\mathcal{K}_j}$ leads to the minimum system expected cost as $T \to \infty$. Let a and b be a pair of strategies such that $\lambda_a \leq \lambda_b$. If $\sum_{i\in\mathcal{K}_a} \varphi_i \leq \sum_{i\in\mathcal{K}_b} \varphi_i$, it is clear that a is optimal regardless of the value of T. Otherwise, equation 6.17 provides the range of values of T such that $Z^{\mathcal{K}_b} \leq Z^{\mathcal{K}_a}$. Notice that this equation bounds T from above, and therefore and therefore a is optimal as $T \to \infty$.

$$T \leq \frac{\sum_{i\in\mathcal{K}_a} \varphi_i - \sum_{i\in\mathcal{K}_b} \varphi_i}{\lambda_b - \lambda_a} \tag{6.17}$$

Corollary: For every network and strategy size, there exists a value T_{crit} such that the optimal solution to the IBSO assignment problem of K probes, $\mathcal{K}^*$, is equal to $\mathcal{K}_E$. Notice that $\sum_{i\in\mathcal{K}_j} \varphi_i \geq \sum_{i\in\mathcal{K}_E} \varphi_i\ \forall\ \mathcal{K}_j \in \mathbf{K}$, $T_{crit} = T_v^{\mathcal{K}_E}$. In the general case, $T_{crit} \geq T_v^{\mathcal{K}_E}$, and it must satisfy equation 6.18 for every efficient strategy $\mathcal{K}_j$. The former implies that $\mathcal{K}_E$ is not necessarily optimal for every value of $T > T_v^{\mathcal{K}^E}$.

$$T_{crit} \geq \frac{\sum_{i \in \mathcal{K}_E} \varphi_i - \sum_{i \in \mathcal{K}_j} \varphi_i}{\lambda_{\mathcal{K}_j} - \lambda_E} \quad \forall\ \mathcal{K}_j \in \mathbf{K} \tag{6.18}$$

- **Property 3:** Let $\Delta Z^{\mathcal{K}^*} = S - Z^{\mathcal{K}^*}$ denote the benefits of information corresponding to the optimal IBSO assignment utilizing K probes. The marginal benefits of information corresponding to a unit increase in T grow at a constant rate equal to $\rho^0 - \lambda_*$, given by equation 6.19.

$$\frac{\Delta(\Delta Z^{\mathcal{K}^*})}{\Delta T} = \frac{\Delta\left(\rho^0(K+T) - \left(\sum_i \varphi_i + \lambda_* \cdot T\right)\right)}{\Delta T} \tag{6.19}$$

- **Property 4:** Let $\Delta Z^{\mathcal{K}^*}\% = \frac{S - Z^{\mathcal{K}^*}}{S}$ denote the benefits of information as a fraction of the total system expected cost resulting from the IBSO assignment of K probes. The marginal percent benefits of information corresponding to a unit increase in T grow at a diminishing rate given by equation 6.21, obtained from equation 6.20 applying the product rule. Notice that the first term in the numerator of equation 6.21 is greater or equal than K, given that $\varphi_i \geq \rho^0\ \forall\ i$. The second term is at most as large as K, because $\mathcal{K}^*$ is efficient, and therefore $\lambda_* \leq \rho^0$.

$$\frac{\Delta(\Delta Z^{\mathcal{K}^*}\%)}{\Delta T} = \frac{\Delta\left(\left(\rho^0(K+T) - \left(\sum_i \varphi_i + \lambda_* \cdot T\right)\right) \cdot \rho^0(K+T)^{-1}\right)}{\Delta T} \tag{6.20}$$

$$\frac{\Delta(\Delta Z^{\mathcal{K}^*}\%)}{\Delta T} = \frac{\frac{\sum_i \varphi_i}{\rho^0} - K \cdot \frac{\lambda_*}{\rho^0}}{(T+K)^2} \tag{6.21}$$

- **Property 5:** A path P^N such that $c_{min}(P^N) > c_{max}(P^j)$ for some $P^j \in \mathbf{P}$ is not utilized by regular assets or probes under an IBSO assignment paradigm. In this definition, $c_{min}(P^j)$ is the minimum possible cost on path P^j, attained when the cost realizations on all the corresponding links are equal to the lowest value in their probability distribution. Similarly, $c_{max}(P^j)$ denotes the maximum cost which may be observed on path P^j.

Proof: In order to prove that P^N is never utilized by non-probe assets it is enough to consider that regular assets are optimally routed under every perceived state. Given the problem assumptions, the later means that they are assigned to the shortest path under information, which satisfies $\rho^{\tilde{x}} \leq c_{s(\tilde{x})}(P^j)\ \forall\ P^j \in \mathbf{P}$, a condition that P^N would never satisfy by definition.

To show that assets utilized as probes are never optimally routed on P^N, notice that there must exist at least one sub path $F_{k-l} \subset P^N_{s-t}:\ c_{min}(F_{k-l}) > c_{max}(J^j_{k-l})\ \forall\ J^i \in \mathbf{P_{k-l}}$. Collecting information along such sub-path would never lead to its utilization by the regular assets, based on the same reasoning we used to prove the first part of this property. Furthermore, $E[c(F_{k-l})] > E[c(J^i_{k-l})]\ \forall\ J^i \in \mathbf{P_{k-l}}$, which means that the sub path is not considered for the routing of regular assets in the absence of information. The combination of the last two facts implies that monitoring F_{k-l} does not introduce changes in the routes followed by regular assets under any circumstance, and therefore has no value for the system (Section 6.3.1). Any path $P^A = P^J - \{F_{k-l}\} + \{J_{k-l}\}$ provides at least as much information as P^N at a lower cost, and is therefore preferred in an optimal solution.

6.5 Solution approach

The solution of the IBSO assignment problem under a SS probe deployment strategy involving K probes entails finding the K-dimensional set of paths $\mathcal{K}_j \in \mathbf{K}$ which minimizes the system expected cost under information $Z^{\mathcal{K}_j} = \sum_{i \in \mathcal{K}_j} \varphi_i + \lambda_j \cdot T$. Each set $\mathcal{K}_j$ represent a feasible probe deployment strategy under which probes i face expected costs given by φ_i. The cost realizations at all links visited by probes define a set of perceived network states, based on which the adaptive routing decisions for the remaining T system assets are made. The set of paths to be followed by the regular assets under every perceived state, along with the corresponding probabilities, defines the optimal hyperpath $\mathcal{H}$, which has an expected cost of $\lambda_{\mathcal{K}_i}$. The solution to an IBSO assignment problem is given by $\mathcal{K}^*$ and the corresponding hyperpath $\mathcal{H}^*$.

For every strategy one can identify the subset of links $i \in \mathcal{I}$ which are visited by at least one probe. Define $I = |\mathcal{I}|$ as the number of links from which information is collected. The set of perceived network states under strategy $\mathcal{K}_j$, $\mathcal{X}_{\mathcal{K}_j}$ consists of all I-tuples $\widetilde{x} = (s^1(\widetilde{x}), s^2(\widetilde{x}), ..., s^I(\widetilde{x}))$, in which $s^k(\widetilde{x})$ indicates the link state experienced at link k under $\widetilde{x}$. The cardinality of $\mathcal{X}_{\mathcal{K}_j}$ is $X = \prod_{k \in \mathcal{K}_j} S^k$, and the probabilities of its elements are given by $r(\widetilde{x}) = \prod_{k \in \mathcal{K}_j} p^k(\widetilde{x})$. The system cost faced by the non-probe system assets under any perceived state is $\rho^{\widetilde{x}}$, the cost of the corresponding shortest path given the available information. In virtue of this $\lambda_j = \sum_{\widetilde{x} \in \mathcal{X}_{\mathcal{K}_j}} p^{\widetilde{x}} \cdot \rho^{\widetilde{x}}$

The problem solution poses similar challenges to those identified in Chapter 4. Except for the incorporation of flow conservation constraints into the upper level problem and the removal of the corresponding budget constraints, the mathematical formulation presented in Section 6.3 is identical the one proposed for the optimal deployment of sensors following IBSO principles.

Notice that, from the perspective of the problem solution, the distinctive characteristic of the model discussed in this chapter is the requirement to select sets of paths to be monitored, instead of individual links. The solution approach adopted for the numerical implementations makes use of an exogenously provided path set to transform the IBSO assignment problem into a large instance of the optimal sensor deployment problem presented in Chapter 4.

The discussion presented in Section 4.5.1 regarding possible exact solution methodologies is valid in the context of the present problem, as well as the methodological framework presented in Chapter 5, which is adjusted and implemented to the solution of the problems presented in Section 6.6. Section 6.5.1 explains the corresponding procedure, as well as some problem characteristics which prevent a more efficient implementation of the state-partitioning approach. The Tabu search heuristic proposed in Section 5.2 can also be adapted to the solution of the IBSO assignment problem, and section 6.5.2 describes the necessary adjustments.

The procedure presented in this section was adequate for the solution of the problem instances presented in section 6.6, which provide very valuable insights into the problem characteristics and behavior. Based on these results, more efficient exact and heuristic approaches may be developed in future extensions.

6.5.1 Implementing a state-space partitioning approach

The problem presented in this chapter can be solved as an instance of the optimal sensor deployment problem for the support of adaptive system optimum routing decisions, provided that the set of all paths $\mathbf{P}_{s-t}$ connecting the analyzed origin-destination pair is available. When this is the case, the solution of a problem involving the utilization of K assets as probes reduces to finding the combination of K paths leading to the lowest system expected cost. Each of these combinations $\mathcal{K}$ may be transformed into a set of links $\mathcal{I} \subset \mathbf{M}$ utilizing Algorithm 9.

Algorithm 9 Generating set $\mathcal{I}$

```
V_j = 0 ∀ j ∈ M
I = ∅
for all (P_i ∈ K) do
  for all (j ∈ P_i) do
    if (V_j = 0) then
      V_j = 1
      I = I + {j}
```

A link-path incidence matrix could be utilized for the same purpose, but in the present implementation paths are stored as a list of links, and therefore Algorithm 9 provides a more convenient approach. Section 6.5.1.1 describes the procedure used to generate $\mathbf{P}$. Notice that different combinations $\mathcal{K}_i$ may lead to the same set I, as exemplified in figure 6.1. This may be of interest in the design of heuristic solution methodologies.

Once the set I is available, the methodology presented in Chapter 5 can be implemented without any further changes to compute λ_i, given which the computation of $Z^{\mathcal{K}_i}$ is straightforward. Notice however that the size of corresponding problem instance, defined by the number of links covered by the probes, is likely to be very large. The perceived state-space grows exponentially as K increases, and the numerical implementations presented in

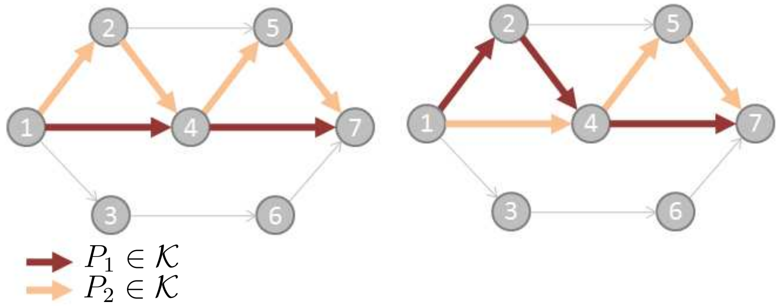

Figure 6.1: Example of duplicated $\mathcal{I}$ sets

section 6.6 suggest that the proposed state partitioning approach may not be sufficient to solve very large problem instances efficiently.

A possible approach to overcome the above mentioned problem is the implementation of Monte Carlo sampling-based **heuristics** (Alexopoulos [1997]). More specialized techniques may be devised by considering that one of the main reasons for the observed performance deterioration is the absence of a deterministic threshold value τ for the shortest expected cost path (Section 5.1.3.2). Given that the deployment of probes may cover a considerable portion of the network, the lack of a finite τ is a likely scenario, and the heuristic generation of a surrogate value may improve the algorithmic performance. Furthermore, depending on how such value is computed, it may be possible to place a bound on the associated error, limiting its impact on the overall solution.

Notice also that the problem characteristics do not allow for a **path-cost based implementation of the state-partitioning** principles. Such approach reduces the number of path states to be accounted for based on the consideration that path costs may take a limited number of values. By collapsing all states exhibiting the same cost into a single super-state, the total number of relevant path states to evaluate may be drastically reduced (Waller and Ziliaskopoulos [2002]).

Figure 6.2 exemplifies the reasons preventing the implementation of such approach in the context of the present problem. In the proposed network, path 2 can take a value of 14 under two possible states, $\widetilde{x} = 1$ and $\widetilde{x} = 2$. The shortest expected costs paths corresponding to these states are $\rho^1 = 14$ and $\rho^2 = 11$, respectively, which are different from each other. If both states are collapsed into a single one, it is not possible to implement the methodology developed in earlier chapters to partition the state-space, given that there is not a single optimal value to compare to the thresholds which define the partitioning rules.

6.5.1.1 Path set generation

For generic directed networks, the enumeration of all $s - t$ paths belongs to the class of #P-Complete problems (Valiant [1979b]), which are counting problems for which there is not a known polynomial time solution algorithm. While the number of such paths may be estimated using Monte Carlo sampling (Roberts and Kroese [2007]), the actual generation of $s - t$ paths for practical implementations is typically accomplished by computing k-shortest paths according to a pre-specified criterion. The unrestricted variant of the k-shortest path problem allows paths to share an unlimited number of links, and can be solved in $O(kn^3)$ using Lalwler's algorithm (Lawler [1976]). Other methodologies impose constraints on the characteristics of the generated paths, such as an upper bound on the number of shared links across paths (van der Zijpp and Catalano [2005]), or on the maximum admissible length (M.Carlyle and K.Wood [2005]). These constraints are typically selected based on the intended use of the generated paths.

While the formerly described techniques are appealing from the perspective of a heuristic solution, the approach taken to solve the examples presented in Section 6.6 required the generation of all $s-t$ paths. In the absence of negative cost cycles, only acyclic paths are of interest. Even though there is s finite number of such paths, it is typically very large, and it grows exponentially with the network size (Korte and Vygen [2000]). In an attempt to reduce the number of paths to consider, a domination criterion was introduced into the path generation process.

The concept of path domination has been utilized by several authors in the implementation of algorithms which implicitly involve path enumeration

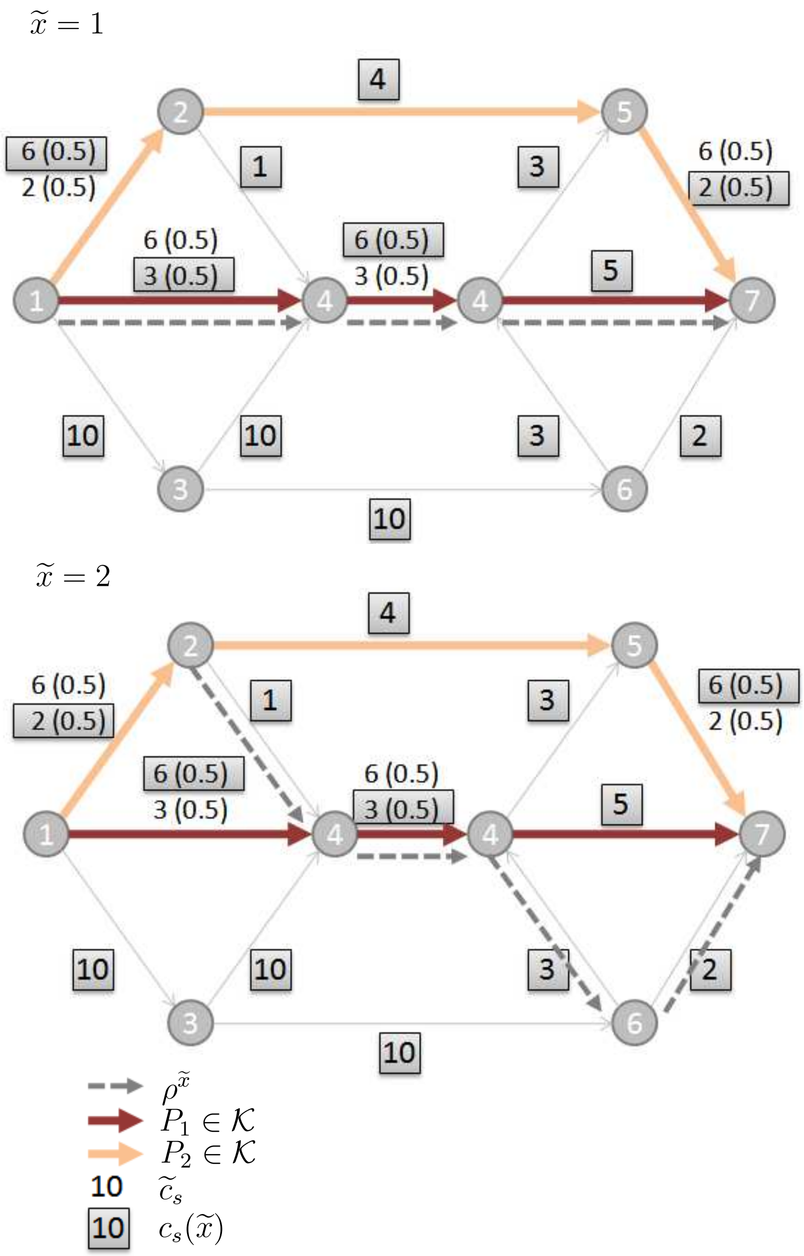

Figure 6.2: Example Network I

(e.g. Miller-Hooks and Mahmassani [1998], Miller-Hooks and Yang [2005]). A domination criterion is basically a set of rules in virtue of which $s-t$ paths can be compared, and eventually discarded, based on conditions specific to the corresponding optimization problem.

For the IBSO assignment property, equation 6.22 presents the criteria utilized to decide whether an $s-i$ path P is dominated, and therefore may be disregarded.

$$\exists\ P_j \in \mathbf{P}:\ l_{s-i}^{P_1} > L_{s-i}^{P_j} \tag{6.22}$$

In this equation, $l_{s-t}^{P_1}$ and $L_{s-t}^{P_j}$ represent the maximum and minimum possible cost on paths P_1 and P_j, respectively, obtained by adding the maximum (minimum) cost realization at every link in the corresponding paths. $\mathbf{P}$ is the set of all paths connecting s and i.

This rule is valid in virtue of Property 5 (Section 6.4) which establishes that a path l_{s-i}^{P} which satisfies equation 6.22 is never used by system assets, and therefore may be disregarded during the problem solution. Notice that if l_{i-j}^{P} represent all the sub-paths in a path l_{s-t}^{k} connecting $s-t$, then l_{s-t}^{P} is non-dominated if and only if all l_{i-j}^{P} are non-dominated. In virtue of this fact, proved below, Algorithm 10 is used to generate the set of all acyclic non-dominated paths in networks with random discrete arc costs.

The proof of the former fact is accomplished in two parts. First we show that if all the sub-paths $F_{ij} \subset P$ are non dominated, then P is non dominated. Let F_{s-i} and F_{i-t} be two of such sub-paths. If they are non-dominated, it means that all sub paths $G_{s-i}$and G_{i-t} are such that $c_{max}(G_{i-t}) \geq c_{min}(F_{i-t})$ and $c_{max}(G_{s-i}) \geq c_{min}(F_{s-i})$. In virtue of this, Equation holds for path $G_{s-t} = G_{s-i} + G_{i-t}$6.23, which implies that P is non dominated.

$$c_{max}(G_{s-t}) = c_{max}(G_{s-t}) + c_{max}(G_{s-t}) \geq c_{min}(F_{s-i}) + c_{min}(F_{i-t}) = c_{min}(P) \tag{6.23}$$

For the second part of the proof, assume that there exists a sub path F_{si} which is dominated by path G_{s-i}. This implies that for some $s-i$ path it is true that $c_{max}(G_{s-i}) > c_{min}(F_{s-i})$. In virtue of this the maximum cost of the path $G_{s-t} = G_{s-i} + F_{i-j}$ is $c_{min}(G_{s-t}) = c_{max}(G_{s-i}) + c_{max}(F_{i-t}) > c_{max}(F_{s-i}) + c_{max}(F_{i-t}) = c_{max}(P)$ which contradicts the initial assumption that P is non dominated and completes the proof.

The algorithm works based on the same principles guiding a typical label-correcting procedure (Ahuja et al. [2002]), and has has an exponential worst case complexity. Notice that in acyclic networks the set of directed paths can be found in $O(m)$ utilizing the same algorithm which provides the topological order (Ahuja et al. [2002]).

Algorithm 10 Non Dominated Path Generation

```
T = {s}
U_i = V_i = 0 ∀ i ∈ N
while (T ≠ ∅) do
  Select i ∈ T
  for all (ij ∈ M) do
    for all (d_k ∈ D_i : k > U_i) do
      if (j ∉ d_k) then {Check Cycle}
        if (l^{d_k}_{s-i} + c^{ij}_{min} ≤ L^{d_m}_{s-j} ∀ d_m ∈ D_j) then {Check Domination}
          if (V_j = 1) then
            U_j = Size(D_j)
          Add d_k + ij to D_j
          Check new dominated paths in D_j
          T = T + {j}
    T = T − {i}
    V(i) = 1
```

For every node $i \in \mathbf{N}$, a different label $d_k \in \mathbf{D}_i$ is used to store each of the non-dominated acyclic $s - i$ paths. The list $\mathbf{T}$ stores all the network nodes which successors need to be re-labeled, and it is initialized with the origin node. At every iteration, all labels in $\mathbf{D}_j$, which represent a possible way to reach i from s, are used to add a new label to the sets $\mathbf{D}_j$corresponding to all nodes emanating form i, provided that no cycle is generated and that the corresponding path is non-dominated. When a new label is added to a node, the same is incorporated to $\mathbf{T}$ in order to update the labels of its successors. Variable U_i is used to keep track of the labels in $\mathbf{D}_i$ which have already been used to generate new paths. V_i allows to use a dequeue implementation (Ahuja et al. [2002]) for the management of $\mathbf{T}$. At termination, the set $\mathbf{D}_t$ is list of all acyclic non-dominated $s-t$ paths. This is guaranteed because nodes enter $\mathbf{T}$ when a new way to reach them has been found, and are removed from $\mathbf{T}$ after updating the possible ways to reach their successors. If $\mathbf{T}$ is empty, then it means that all possible ways to reach every node have been enumerated.

Notice that the set of non-dominated paths depends on the maximum and minimum values at the network links. These change during the evaluation of a probe deployment strategy. Given that $\mu_j \geq \min(c^j)$, the paths found to be dominated implementing Algorithm 10 remain dominated. However, given equation 6.22, additional paths may become dominated, which may be used as the basis for path-based heuristic procedures in further implementations.

6.5.2 Possible adjustments to the Tabu search procedure

The Tabu search procedure introduced in Section 5.2 can be implemented to the solution of the presented problem following a similar procedure to that described in Section 6.5.1, and assuming that the set of non-dominated network paths is available. The methodology can be easily adjusted to select sets of paths instead of links, while the corresponding penalties and adaptive memory structures may still be defined on a link basis (Equations 6.24 to 6.26).

$$\text{Tabu\_R}(P_k) = \sum_{j \in P_k} \text{Tabu\_R}(j) \tag{6.24}$$

$$\text{Tabu\_F}(P_k) = \sum_{j \in P_k} \text{Tabu\_F}(j) \tag{6.25}$$

$$\delta_{max}(P_k) = \sum_{j \in P_k} \delta_{max}(j) \tag{6.26}$$

The procedure was tested on the networks analyzed in the following section, but given characteristics of the considered problem instances it did not lead to faster solutions and therefore was not implemented to the numerical analyses. In the presented framework the Tabu methodology is intended to reduce the number of strategies to be evaluated, which are given by combinations of K paths for a K-probe IBSO assignment problem. However, the algorithm provides significant advantages when the total number of strategies to be evaluated is high (typically in the order of 10^5 and above). For smaller instances, the number of evaluations necessary to identify a good solution is usually comparable to the number of evaluations required by an exact solution approach. For the cases analyzed in Section 6.6 the total number of combinations is relatively small ($\sim 10^4$) and therefore it was not necessary to implement the Tabu methodology. Larger problem instances (in terms of the number of assets utilized as probes) were not considered due to the computational effort involved in evaluating each of the corresponding strategies. Section 5.1.6 discussed the limitations of the state-partitioning approach under this circumstances. The insights provided by the exact problem solution may be used to devise more efficient Tabu search procedures.

6.6 Implementation: Assessing the performance of Information-Based System Optimum assignment strategies

This section presents a detailed analysis of the performance of Information-Based System Optimum (IBSO) assignment strategies on two different example networks, assuming a Serial-Sequential (SS) probe deployment approach. The methodology described in Section 6.4 was implemented to the analysis of Networks I and II, already introduced in Section 5. Section 6.6.1 briefly discusses the performance of the solution approach, and suggests desirable extensions and refinements. The results are discussed in the following sections, which analyzes the impacts of IBSO assignment under SS probe deployment from different perspectives. One of the principal performance measures utilized in this section is the difference between the system expected cost under a K-probes IBSO assignment strategy (Z^K) and

S, the expected cost of a traditional System Optimum (SO) deployment. Such difference is denoted ΔZ^K, and $\Delta Z^K\% = \frac{S-Z^K}{S}$ is the corresponding percent value. Section 6.6.4 analyzes these parameters as a function of T, the number of system assets which are not utilized as probes.

Another important characteristic of an IBSO assignment discussed in this chapter is the difference between the expected costs faced by the probes (φ_i) and λ, the cost faced by each of the remaining system assets. The parameter $\varepsilon = \frac{\varphi_{max}}{\lambda} - 1$, where $\varphi_{max} = \max_i \varphi_i$, measures such difference, and is an indication of the value of information and the "unfairness" of the deployment strategy. The later is also reflected by $\Delta\varphi_{max} = \frac{\varphi_{max}}{\rho^0} - 1$, which represents the deterioration in the expected cost faced by the probes with respect to a SO assignment. Conversely, $\Delta\lambda = 1 - \frac{\lambda}{\rho^0}$ measures the improvement with respect to a SO approach experienced by the remaining system assets. In both cases, ρ^0 is the value of the shortest expected cost path when no information is provided, which is the cost paid by all assets under a SO deployment under the problem assumptions. A comparable value under IBSO assignment is $\nu = \frac{Z^K}{W}$, where $W = T + K$ is the total number of assets in the system.

Section 6.6.3.1 discuses some qualitative aspects of the routes utilized under an IBSO deployment approach, comparing them to the optimal paths resulting from SO assignment, and to the solutions obtained in Chapter 5. Notice that the solution to an IBSO problem consists of the set of routes to be followed by the assets utilized as probes, and of the optimal routing solutions for the remaining assets under each perceived network state, given by the corresponding hyperpath.

Similarly to what we observed in Chapter 5, the numeric value of the performance measures described above is likely to vary widely for different networks depending on their topology, cost structure and corresponding probability distributions. The goal of this section is to improve our intuitive understanding of the nature of the effects of an IBSO assignment approach, and how it leads to a different system behavior than a SO deployment strategy. This is critical in order to identify practical implementations which would benefit the most from the proposed approach, and to develop refined models able to maximize the benefits of information availability.

6.6.1 Algorithmic performance

Tables 6.1 and 6.2 summarize the algorithmic performance on Networks I and II for different numbers of probe vehicles. The networks account for 46 and 187 non dominated acyclic paths, respectively. Path lengths range between 3 and 8 links in Network 1, and between 3 and 13 links in Network II, which leads to considerably large problem instances, with state-spaces cardinalities easily reaching the order of 10^6.

Notice that the total numbers of acyclic paths on Networks I and II are 68 and 720, respectively. The utilization implementation of the path-domination criteria presented in Section 6.5.1.1 allowed to reduced the number of paths to be considered .

Performance measures are analyzed by deployment strategy type, which are defined in Section 5.2.1 based on the overlap between the corresponding probe routes $\mathcal{P}_i$ and the shortest expected cost path when no information is provided, $\mathcal{L}^0$. The reported values are the average performance across all the possible strategies in each category given by $|\mathbf{K}^T|$, where $\mathbf{K}^T$ is set of all strategies of type $T = \mathrm{I}, \mathrm{II}, \mathrm{III}$. Notice that strategies are defined in terms of paths, and therefore strategies Type II, which include exclusively links in $\mathcal{L}^0$, are only possible in one-probe cases.

Performance is measured based on the percentage of total states evaluated, $\%_{eval} = \frac{\text{Evaluations}}{|\mathcal{X}^{\mathcal{K}_i}|}$, where $|\mathcal{X}^{\mathcal{K}_i}|$ is the state space containing all possible states perceived under probe deployment strategy $\mathcal{K}_i \in \mathbf{K}$.

The results exhibit similar trends to those observed in Section 5.1.5, suggesting a much better performance on Network II in virtue of its topological characteristics. The performance is similar for the two strategy sizes considered for Network II, which indicates the same type of stable behavior noticed in the earlier numerical analyses. This is very encouraging, given the considerably large size of $\mathcal{X}^{\mathcal{K}_i}$. The maximum number of simultaneous partitions P is fairly low, but higher than the corresponding

	1 Probe			2 Probes		
	Type III	Type I	Type II	Type III	Type I	Type II
$\lvert\mathbf{K}^T\rvert$	28	17	1	899	136	-
$\lvert\mathcal{X}^{\mathcal{K}_i}\rvert$	1325.1	3386.5	27	224009.6	204692.1	-
$\%_{eval}\mathcal{X}^{\mathcal{K}_i}$	89%	33.3%	88.9%	89.7%	42.9%	-
$\mathbf{P}$	0.5	1.9	1	2.0	3.4	-

Table 6.1: State-space partitioning algorithm performance in Network I

	1 Probe		
	Type III	Type I	Type II
$\lvert\mathbf{K}^T\rvert$	86	100	1
$\lvert\mathcal{X}^{\mathcal{K}_i}\rvert$	2283.5	2088.2	12
$\%_{eval}\mathcal{X}^{\mathcal{K}_i}$	50.0%	13.2%	50.0%
$\mathbf{P}$	2.0	1.8	2

Table 6.2: State-Space partitioning algorithm performance in Network II

values for smaller state-spaces.

Even though the algorithmic performance is satisfactory for the analyzed cases, it is important to notice that the number of possible perceived states grows extremely fast as larger deployment strategies are considered. Under such circumstances, the number of evaluations required by the proposed approach may be prohibitive, even if it represents a small percentage of the state-space size. Heuristic approaches based on Monte Carlo sampling, such as those presented in Alexopoulos [1997], may be a feasible approach to overcome this problem. Other bounded-heuristic procedures may be devised based on the problem properties, as discussed in Section 7.3.

6.6.2 System expected cost

The system expected cost under an IBSO approach is given by the summation of the expected cost faced by the assets utilized as probes and the cost faced by the remaining system assets. Tables 6.6 and 6.4 present different measures related to the system expected cost under IBSO assignment for

	1 Probe $T_{crit} = 14$			2 Probes $T_{crit} = 19$		
	$T = 1$	$T = 35$	$T = 550$	$T = 1$	$T = 15$	$T = 20$
S	304.4	5479.2	83862.2	456.6	2587.4	3348.4
Z	**303.3**	**5440.5**	**82796.4**	**492.8**	**2526.6**	**3227.8**
$\Delta Z\%$	0.4%	0.7%	1.3%	-8%	2%	4%
v	151.6	151.1	150.3	164.3	148.6	146.7
φ_i	152.2	193.5	286.4	345.7	415.8	468.6
φ_{max}	152.2	193.5	286.4	152.2	217.1	275.1
λ	151.1	149.9	149.7	147.3	140.7	138.0
$\Delta\varphi$	0%	27.1%	88.2%	127%	173%	208%
ε	7%	29.1%	91.3%	135%	195%	240%

Table 6.3: Results summary for Network I

Networks I and II respectively, and for different values of T. For Network I two different strategy sizes are considered. Additionally, Table 6.5 presents similar information for Network I under the two-state probability distribution case described in ().

The reductions in system expected cost with respect to a SO assignment approach, $\Delta Z\%$, range between -8% and 6% in Network I. Even though the value of such impacts depends on the considered implementation, the qualitative analysis of the behavior of $\Delta Z\%$ provides very interesting insights. The first important observation is that the number of assets not used as probes, T, plays a fundamental role on the magnitude, and even the direction, of the perceived impacts. This is expected, given that the additional cost paid by the probes in order to collect information is compensated only by the gains experienced by the remaining assets in the system. The negative value observed under the 2-probe deployment strategy for Network I when $T = 1$ reflects the fact that $T_v^{\mathcal{K}_i} > 1$ for all strategies $\mathcal{K}_i$of size 2 (Section 6.4), and therefore utilizing 2 vehicles as probes does not benefit the system. Notice that the negative impact on the system is a result of forcing the model to select different routes for every utilized probe. The actual optimal strategy would route both probes on the same path selected for the strategy of size 1.

1 Probe

	$T_{crit} = 4$	
	$T = 1$	$T = 15$
S	129	1032
Z	**125.9**	**968.1**
$\Delta Z\%$	2%	6%
v	63.0	60.50
φ_i	64.5	71.2
φ_{max}	64.5	71.2
λ	61.4	59.8
$\Delta\varphi$	0%	4%
ε	5%	19%

Table 6.4: Results summary for Network II

Section 6.6.4 will further discuss the effect of T on the system expected cost.

The expected cost paid by the probes $\varphi_i$is always equal or higher than the cost faced by the remaining system assets. The later are optimally routed based on the findings of the probes, and therefore they never face higher expected costs (nevertheless, the actual costs paid by these assets may be higher than expected, depending on the realizations observed at those links which remain uncertain). The observed values of $\Delta\varphi$ range between 0%, when the probe is assigned to $\mathcal{L}^0$, and 208%. Larger values occur when T is higher, and therefore more assets may benefit from the finding of those utilized as probes. The values of ε, which range between 5% and 240%, reflect the same behavior. This may have important practical implications, given that probes are paying considerably higher costs than the remaining assets. For certain implementations, decision makers may want to limit either ε or $\Delta\varphi$, depending on whether they intend to limit the total cost faced by the probes or the inequity between system assets.

The results suggest that average expected cost per asset v decreases a function of the number of deployed probes as long as $T > T_{crit}$. However, this may not be the case for lower values of T. In Modified Network I for example, the utilization of three assets as probes when $T = 15$ yields lower benefits

	1 Probe	2 Probes	3 Probes
	$T_{crit} = 14$	$T_{crit} = 19$	$T_{crit} = 45$
	$T = 15$	$T = 15$	$T = 15$
S	2672	2839	3006
Z	2490.4	2525.7	2741.9
$\Delta Z\%$	6.80%	11.0%	8.9%
v	156	149	152
φ_i	186	366	587
φ_{max}	186	186	221
λ	151.3	144.0	143.7
$\Delta\varphi$	11.4%	119.2%	251.5%
ε	23.1%	156.7%	311.6%

Table 6.5: Results summary for Modified Network I

than the implementation of a 2 probe strategy (9% and 11%, respectively). One may consider that the value of $\Delta\varphi$ for $K = 1$ is an indicator of the value of information for the considered system, given that it represents the maximum additional cost that the system can afford to paid in order to collect information.

The results in Network II support the trends described in the previous paragraphs. It is interesting to notice that for this network the value of $\Delta Z\%$, which ranges between 2% and 6% is comparable to the results reported for Network I, and is achieved at a much lower cost, with φ_i ranging between 0% and 4%. This illustrates the degree of dependence of the costs and benefits of information on network characteristics, and motivates the search for models tailored to different applications.

Some additional tests were conducted in order to assess the effects of selecting a suboptimal strategy for a given value of T. In Network I for $T = 1$ and $K = 1$, the system cost obtained by deploying the probe asset on the optimal path corresponding to $T = 15$ is 38 % higher than the optimal system expected cost, and actually higher than S. However, if the optimal strategy for $T = 1$ is utilized for larger values of T the impacts are less extreme, ranging between 1% and 5% for $T = 15$ and $T = 100$, respectively. These results are also important from a methodological perspective, and they may be used to guide heuristic solution approaches tailored to each particular implementation

6.6.3 Optimal routing strategies

Figure 6.3 displays the optimal probe routes ($\mathcal{P}$) for $K = 1$ and values of T above and below T_{crit}, along with the links included in the corresponding hyperpaths ($\mathcal{H}$). Figure 6.4 presents the same information for Network II.

Notice the paths corresponding to different values of T do not necessarily overlap, suggesting that completely different routing strategies may be appropriate based on the system size.

It is also interesting to observe that the links in $\mathcal{H}$ cover a considerable portion of Network I, and a very specific section of Network II. This suggests that the use of information leads to a larger utilization of the available resources. Some paths or network sections which are not part of a solution under uncertainty may be utilized under an IBSO deployment strategy. The former suggests an alternative utilization of the presented models to promote a more efficient network utilization. The proposed framework can be used to identify which information is be relevant in order to "activate" specific network links, giving them a positive probability of being used.

Figure 6.5 displays the results corresponding to a two probe deployment strategy on Network I, which exhibit similar trends. Similarly to what we observed in Chapter 5, the optimal solution for a strategy of size 2 does not necessarily include the optimal probe route identified for $K = 1$.

The displayed results also show a trend to include the shortest expected cost path $\mathcal{L}^0$ in the set of optimal probe routes for low values of T, even though it is usually not part of the optimal strategy for $T \geq T_{crit}$.

6.6.3.1 Comparing probe routes to the optimal location of static sensors

The comparison of the links in $\mathcal{P}^*$ and the optimal links to be monitored based on the models presented in Chapter 5 leads to some interesting observations.

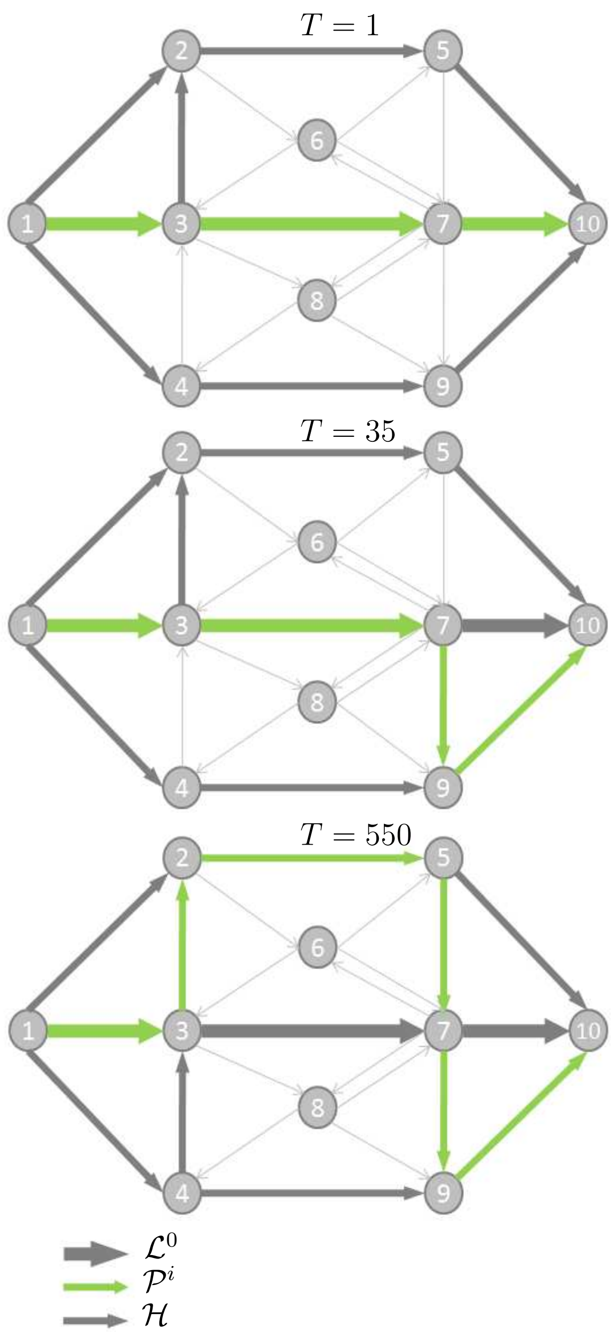

Figure 6.3: Optimal routing strategies using one probe in Network I ($T^1_{crit} = 15$)

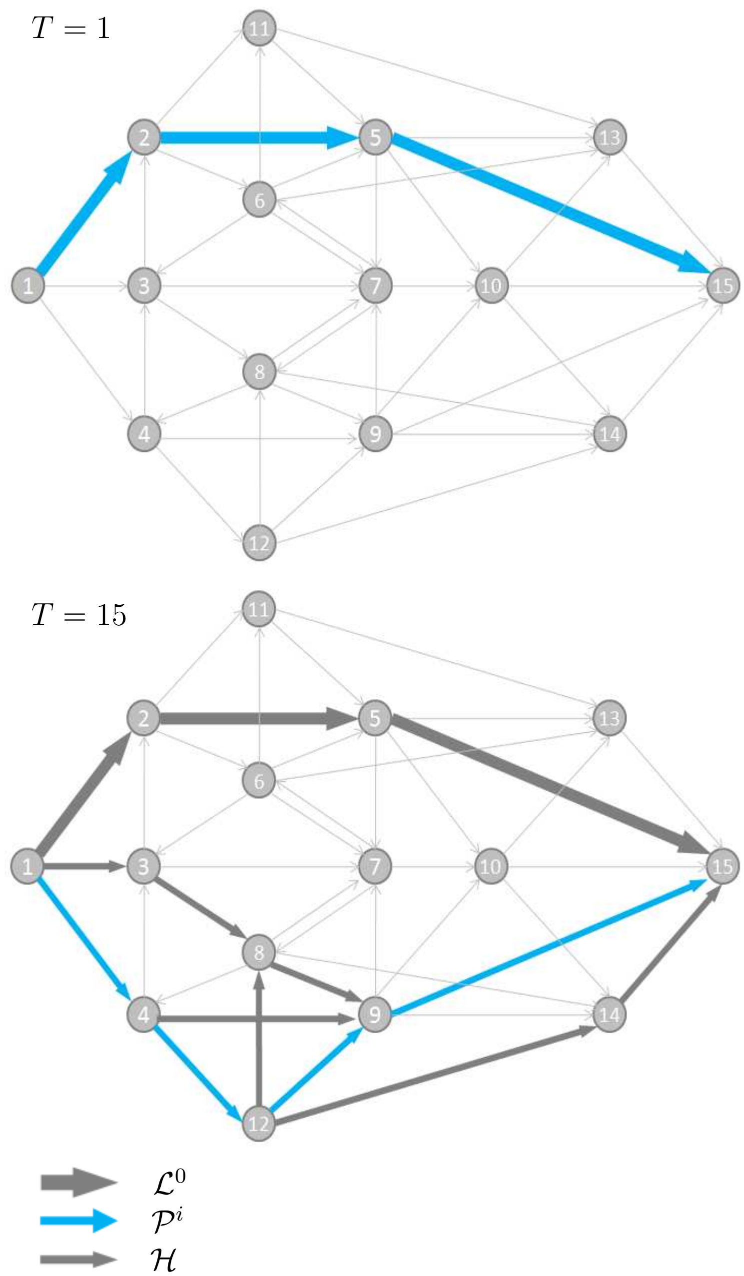

Figure 6.4: Optimal routing strategies using one probe in Network II ($T_{crit}^1 = 4$)

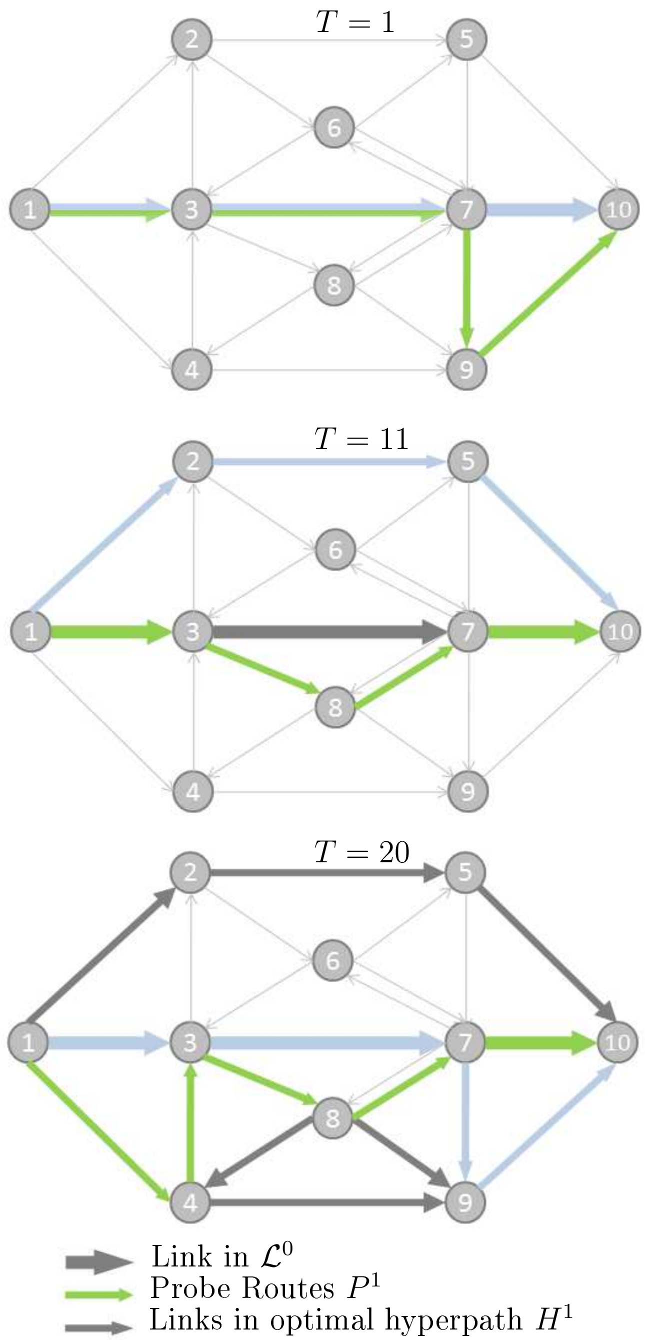

Figure 6.5: Optimal routing strategies using two probes in Network I ($T^1_{crit} = 20$)

The overlap between the results obtained from the two solution approaches (in terms of measured links) seems to be highly dependent on the network characteristics. For Network I, only 3 of the links monitored under an optimal 6-sensor deployment strategy are part of the optimal probe path under a one-probe IBSO assignment approach. In Network II, all the links covered by the probe route are also monitored based on the models for static sensor deployment previously discussed.

In general, the observed differences in terms of covered links reflects the fact the models analyzed in this chapter associate a cost to the acquisition of information, whereas the data collected with sensors deployed according to the models in Chapter 5 is “free” from the system perspective. The vehicles used as probes need to reach the links with the most valuable information, which may impose additional costs that are not compensated by the corresponding benefits, even for large values of T. This clearly depends on the network topology and costs distributions. In Network II the assets are able to cover the most beneficial links at a relatively low cost, while in Network I only those critical links closer to the origin and destination are monitored using probes.

The impacts of considering the cost of information are reflected on the corresponding values of λ. These can be compared to the system expected cost under a k-sensor deployment strategy, where k corresponds to the length of the path utilized by the probe. In the case of Network II, both values are equal, reflecting that the maximum benefits attainable by monitoring four links are achieved by deploying one probe. For Network I, λ is 1.1% higher than the cost experienced by the system under a 5 sensor deployment strategy. Notice that models presented in Chapter 5 to find the optimal deployment strategy of K sensors provide a lower bound to the system benefits obtained by deploying a probe along a path of length K. Under the former assumption, the model described in this chapter may be regarded as a more constrained version of the models in Chapter 5, which forces to deploy available sensors along a path connecting origin and destination.

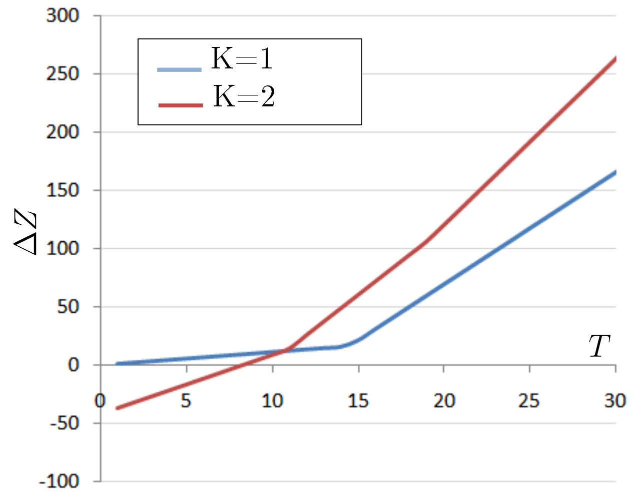

Figure 6.6: Benefits of information for optimal probe routing strategies on Network 1

6.6.4 Impacts of the system size on the attainable benefits of information

The analysis conducted in section 6.6.2 illustrates the importance of the system size on the maximum benefits which may be attained by implementing IBSO deployment strategies. This is a direct consequence of properties 1 and 2, presented in Section 6.4. Figures 6.6 presents the benefits of information as a function of T in Network I for strategies of size 1 and 2.

As expected based on property 3, the benefits of information increase at a linear rate as a function of T. The rate of increase, which is a direct function of the cost paid by the non- probe assets, changes based on the optimal routing strategy at T. The maximum rate is achieved at T_{crit} (Section 6.4). Figure 6.7 illustrates the fact that, as $T \to \infty$ the strategy providing the highest increase rate dominates the remaining ones, even when this may not be the case for reduced values of the variable. This behavior is of interest and it may be crucial in the identification of solution methodologies tailored to specific applications. For implementations involving a large number of assets, the cost paid by the

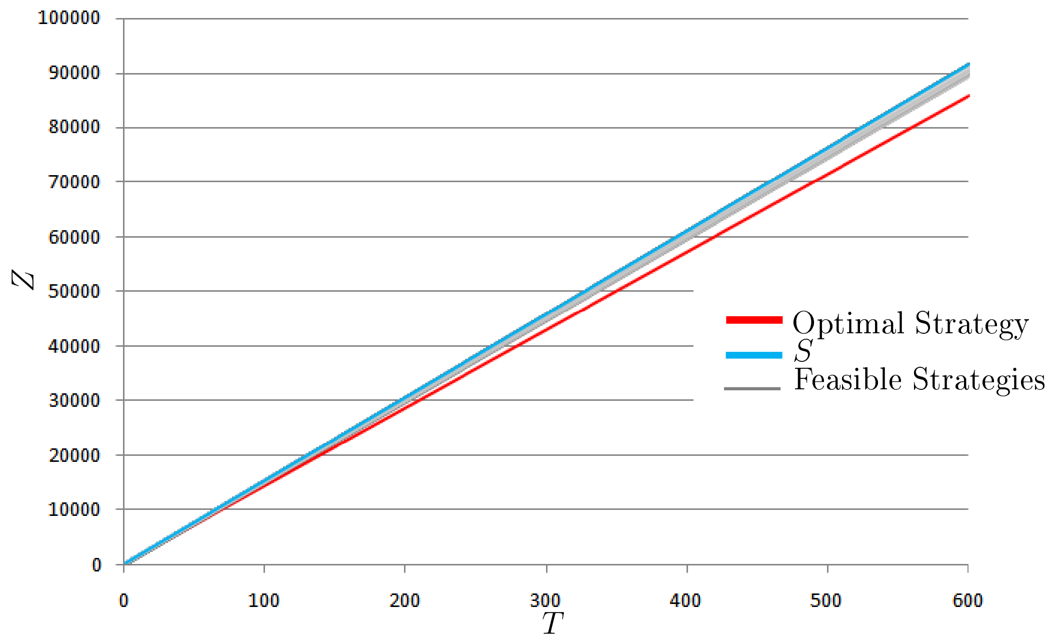

Figure 6.7: System expected cost under various one-probe strategies (Network I)

probes may be neglected, and the solution methodology may be focused on finding the strategy leading to the minimum λ. However, for applications that account for small values of T the cost paid by the assets utilized as probes plays a fundamental role, and ε can be considered as an indicator of the value of information in the corresponding network.

Figures 6.8 and 6.9 present the value of $\Delta Z\%$ as a function of T for Networks I and Modified Network I (Table A.6), respectively. In this figures it is possible to observe the discontinuities introduced by the changes in λ, and the diminishing marginal value of information, reflected in the progressive flattening of the corresponding curves. Additionally, the plots suggest that for sufficiently large values of T strategies utilizing a larger number of probes lead to larger benefits.

To complement the previous plots, table 6.6 displays the value of T_v^i for every 1-probe deployment strategy considered for Network I, which correspond to all the non-dominated acyclic paths in such network. Notice that the

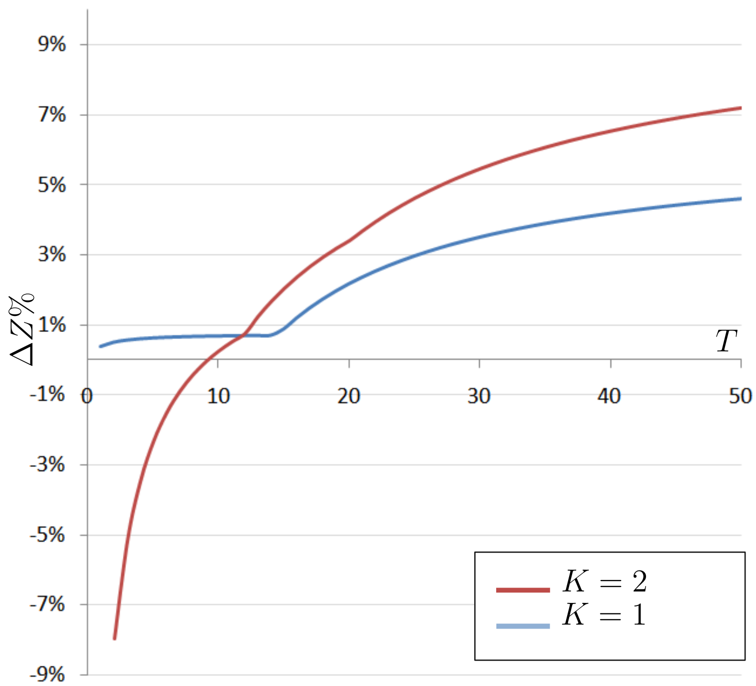

Figure 6.8: Percent benefits of information for optimal probe routing strategies on Network 1

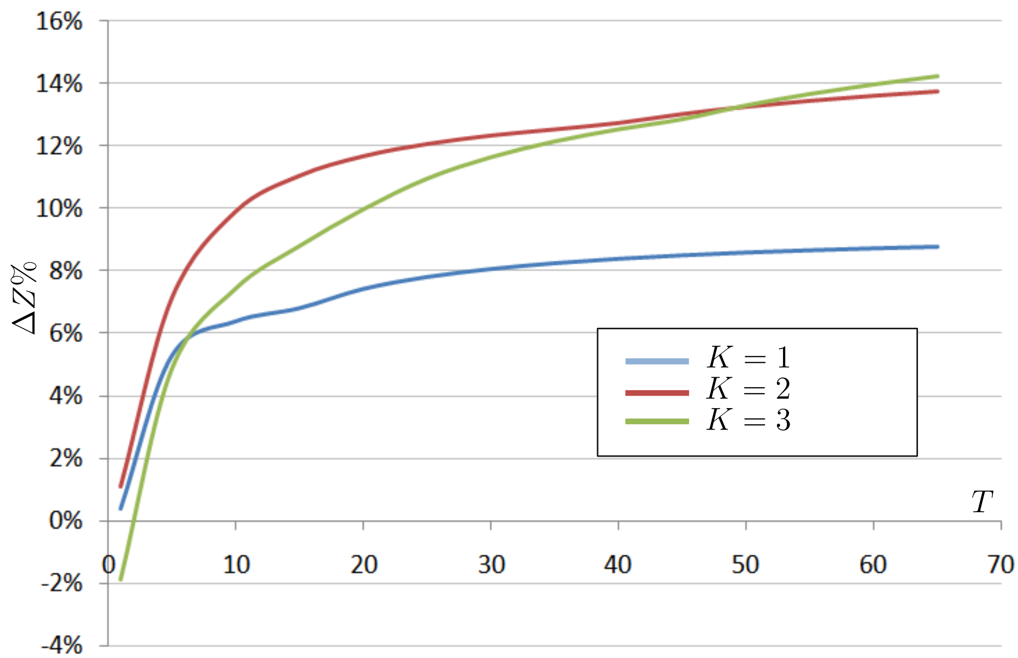

Figure 6.9: Percent benefits of information for optimal probe routing strategies on modified Network 1

shortest expected cost path $\mathcal{L}^0$ (Strategy 7) is a feasible option (i.e. leads to a value of $Z \leq S$) for any value of T ($T_v^7 = 0$). In effect, given that routing a probe on this path does not impose on it any additional cost with respect to a SO assignment solution, the information collected along $\mathcal{L}^0$ is available for "free", and its utilization cannot harm the system performance. Conversely, some paths are not feasible under any value of T ($T_v^i = \infty$). This reflects the fact that the information collected from such paths does not lead to any benefit for the system ($\lambda_{\mathcal{K}_i} \geq \rho^0$), and therefore the additional cost paid to collect it is never compensated.

It is also interesting to notice that $T_{crit} \geq T_v^{\mathcal{K}^{min}}$, where $\mathcal{K}^{min}$ denotes the strategy leading to the lowest value of $\lambda_{\mathcal{K}_i}$. In this example strategy 20, which presents the lowest value of $\lambda_{\mathcal{K}_i}$, is feasible for $T \geq 13$ but it does not become optimal until $T = T_{crit} = 16$. This result reflects property 2, and implies that even for large values of T neglecting of the cost faced by the assets utilized as probes may lead to a suboptimal (but nonetheless acceptable) solution.

	λ	$\sum_i \varphi_i$	T_{crit}
1	152.2	217.1	∞
2	150.9	222.9	53
3	151.6	299.3	241
4	152.2	266.0	∞
5	151.8	271.8	292
6	152.2	348.2	∞
7	151.1	152.2	0
8	152.2	228.6	∞
9	152.2	239.3	∞
10	150.3	245.1	48
11	151.4	321.5	218
12	152.2	231.7	∞
13	151.0	237.5	69
14	152.2	313.9	∞
15	152.2	288.2	∞
16	151.1	294.0	126
17	152.2	370.4	∞
18	151.7	289.7	295
19	151.1	198.7	41
20	152.2	275.1	∞
21	151.8	336.2	449
22	151.3	195.4	49
23	149.9	193.5	18
24	150.5	269.9	67
25	150.8	280.6	90
26	149.7	286.4	55
27	150.5	362.8	124
28	150.8	273.0	84
29	149.9	278.8	54
30	150.5	355.2	119
31	150.8	329.5	124
32	150.0	335.3	83
33	150.5	411.7	153
34	150.5	201.5	29
35	150.9	277.9	99
36	150.4	236.2	47
37	150.9	312.6	122
38	151.4	323.3	207
39	150.3	329.1	91
40	151.0	405.5	203
41	151.4	315.7	198
42	150.4	321.5	92
43	151.0	397.9	197
44	150.5	378.0	133
45	149.9	240.0	39
46	150.4	316.4	94

Table 6.6: T^1_{crit} for all feasible one-probe strategies on Network I

6.7 Summary

This chapter introduces the concept if Information Based System Optimum (IBSO) assignment, proposes a mathematical formulation capturing the novel paradigm, and presents numerical results illustrating the problem properties and main methodological challenges.

The IBSO assignment problem extends the basic principles of cooperative assignment observed in a traditional System Optimum approach in order to account for the collection and utilization of information in stochastic networks (Section 6.2). It considers the utilization of a subset of system assets as probes, used to collect information regarding the cost realizations throughout the network. Probes may face higher expected costs than the remaining assets, but in an optimal assignment these are compensated by the benefits accrued by the system. The new modeling framework has an enormous potential to reduce the negative impacts of uncertainty on the solution of a number of problems on stochastic networks.

The problem presents interesting properties, describe in Section 6.4. Some of these actually may be used to define parameters to characterize a system, such as the minimum number of assets required to justify the utilization of a given number of probes. The mathematical models described in Section 6.3 are able capture the tradeoffs between the value of information and the additional cost faced by the probes in order to collect it. Their exact solution is challenging, given their combinatorial nature (Section 6.5). A solution methodology based on state-space partitioning was implemented to the analysis of several problem instances (Section 6.6). The procedure has an exponential complexity, and a it requires the enumeration of all possible paths connecting the analyzed origin-destination path. A path domination criterion was defined and implemented in order to alleviate the computational effort.

The qualitative analysis of the numerical results illustrates the problem properties and provides valuable insights into the potential modeling improvements and practical applications. Expected cost reductions ranging between 2% and 6% were measured on the analyzed systems. More importantly, the new assignment paradigm was found to lead to considerably different routing decisions than a naive approach, which actually depends not only on the network characteristics but on the system size. The marginal benefits of information grow at a constant rate with the system size, while the gains expressed as a fraction of the system cost exhibit a decreasing rate of marginal growth. For a sufficiently large system, the cost faced by the probes was found to be irrelevant, which is expected, and the optimal probe assignment strategy is that leading to the lower system expected cost. However, in many potential applications (Section 7.2) the system size is may be relatively small, and modeling the cost of information acquisition becomes crucial. The results suggest that the provision of information leads to a more efficient utilization of the system, given that paths which were not considered under the expected-cost based routing may become appealing given the information revealed by the probes. This suggests an alternative implementation of the proposed models to understand what information would be necessary in order to promote the utilization of specific network links.

The model discussed in this chapter provides a flexible tool to measure and understand the benefits of information in the context of adaptive system optimum assignment strategies. Based on the findings presented here, practical implementations may be devised, along with the corresponding efficient solution methodologies. This is an important step towards a more efficient utilization of information in the optimization of transportation systems.

Chapter 7

Conclusions, Applications, and Extensions

Traffic information, now available through a number of sources, is re-shaping the way planners, operators and users think about the transportation network. It provides a powerful tool to mitigate the negative impacts of uncertainty, and an invaluable resource to manage incidents and other causes of non-recurrent congestion. Information also invites to think about traditional transportation problems from a different perspective which may take advantage of the improved understanding of the network state.

This dissertation proposes a novel system-optimum assignment paradigm which takes into account the ability of assets to collect information as they travel through the network. It also presents a methodology to design information collection strategies based on the impacts of such data on system-optimum routing decisions, an approach not considered in the existing literature. Specialized exact and heuristic solution techniques were developed based on the problem properties, and implemented to the analysis of several example problems. The conducted numerical analyses suggest that the models introduced in this work provide a means to utilize information for improving system performance. The results illustrate interesting problem properties, which point to possible practical implementations, and have important methodological implications. Furthermore, some of the studied properties define network parameters which may be used to characterize the susceptibility of systems under uncertainty to benefit from information provision. The models proposed in this work constitute an initial step towards enhancing information collection and utilization strategies. Based on the findings presented here, a number of applications may be envisioned, which efficient solution could promote a more effective utilization of existing and upcoming technologies, fostering the full realization of their potential benefits. The following sections summarize and integrate the research conducted for this dissertation, and suggest further research directions.

7.1 Optimal sensor deployment for system-optimum adaptive routing support

The optimal sensor deployment model presented in this work contributes to the existing literature on efficient collection of information from static sensors. The review conducted in Chapter 3 suggests that major improvements are still possible on such field, particularly if new paradigms for the utilization of information become available. Existing approaches in the area focus on improving system-monitoring capabilities, but hardly consider the impacts of the collected information on the system performance, or in adaptive routing decisions.

The deployment model proposed in Chapter 4 identifies the optimal location of a fixed number of static sensors in a network with stochastic arc costs, in such way that the expected cost faced by a set of optimally routed assets is minimized. The information provided by the sensors generates a set of perceived network states, based on which the optimal paths to be followed by the system assets may be adapted. Such routes correspond to the shortest expected cost path under each information set given the problem assumptions. The model is suitable for a number of interesting applications, ranging from the deployment of sensors during rescue operations, to data filtering for online routing purposes The bi-level stochastic program used to formulate the problem (Section 4.2) provides valuable insights into the problem properties, and is used to derive an expression for the marginal value of information, which is proved to be always non negative.

Numerical analyses, conducted implementing the specialized methodology described in Section 7.3, illustrate how the utilization of information to adjust system optimum routing strategies may improve the system perform ace. Gains of up to 4% with respect to a no-information scenario were measured. The absolute value of the accrued benefits is likely to vary widely depending on the characteristics of specific networks, and the practical value of the observed improvements depends on the considered application. However, it is important to notice the solution obtained using the novel approach was up to 50% more effective than a deployment strategy based only on maximum link variance, which is promising. The qualitative analysis of the solutions also reveals interesting problem characteristics, such as a "synergic" behavior, in virtue of which the benefits obtained by jointly monitoring a group of links may be greater than the improvements accrued by placing sensors on any subset of such group. Additionally, the consideration of the links involved in the optimal hyperpaths suggests that the availability of more information eventually leads to the utilization of a larger set of paths. The latter points to an alternative application of the proposed models, which may be implemented to identify information provision patterns that promote the usage of specific network links.

There are many potential applications for the methodologies presented in this section. The deployed sensors need not be traffic sensors, but may be special instruments used to measure the terrain conditions in areas affected by a natural disaster, such as earthquakes or floods. The devises can be optimally placed after the occurrence of a specific incident in order to identify feasible routes for the emergency vehicles or the evacuation of victims.

Another application may be envisioned if one considers that the solution of the optimal sensor deployment strategies identifies the set of links which provide contain the most valuable information. The methodology may be used to improve the utilization of data from sensors already deployed. By identifying those sensors which information is more relevant for a particular routing decision, the models can be used to reduce the amount of data that needs to be processed in order to generate adaptive routing strategies, allowing for faster and more effective solutions. Finally, other potential applications involve problems such that network links represent the duration of the different steps of a process or group of integrated processes, which may be accomplished in various ways, represented by paths. The placement of a sensor is equivalent to monitoring the duration of the corresponding process, and the proposed models may be used to identify which steps play a more fundamental role on the system behavior

7.2 IBSO Assignment

The concept of Information-Based System-Optimum (IBSO) assignment introduced in Chapter 6 is inspired by the equilibrium approaches discussed in the literature review, which aim to capture the system-level impacts of individual adaptive behavior on stochastic networks. However, the IBSO paradigm is fundamentally new, given the assumption that the information used to adjust the system's routing decisions is collected by a fixed subset of the assigned assets, which are utilized as probes. The selection of the paths followed by the probes takes into account the value of the information collected along them in addition to the corresponding expected cost. As a consequence, assets utilized as probes may face higher expected costs than regular system assets, which are optimally routed under every possible state revealed by the probes.

The problem is formulated as a bi-level stochastic program, which assumes flow-independent link costs and a serial-sequential probe deployment approach, such that all the probes enter the system together and before the regular assets. The proposed model is such that the marginal value associated to the utilization of additional assets as probes may be negative. The later reflects the fact that the cost involved in acquiring information may not be compensated by the system-level benefits. The problem presents interesting properties, some of which may be used to define parameters that characterize the system under study, such as the minimum number of regular assets which justifies the utilization of s probes as assets. Defining this type of properties contributes to a better understanding of specific instances of the studied problem and the underlying network. They allow measuring how susceptible a network/problem is to benefit from information, how costly it is to achieve a minimum level of improvement, what is the maximum gain that may be expected from utilizing some assets as probes, among other important characteristics. Theoretical properties may also have important methodological implications. For example, the fact that the benefits of information grow linearly with the system size suggests that, for a sufficiently large system, the cost faced by the probes may be disregarded in the search for an optimal solution.

A variation of the methodology developed to solve the optimal sensor deployment problem was used to conduct numerical experiments assessing the performance of the IBSO assignment approach. The qualitative analysis of the corresponding results illustrates the problem properties, and provides valuable indications regarding desirable methodological improvements and potential practical applications. The measured expected cost reductions, with respect to a no-information scenario, ranged between 2% and 6%. More importantly, the new assignment paradigm was found to lead to considerably different routing decisions than a naive approach. The selected paths depend on both, the network characteristics and the system size. Additionally, similarly to what we observed on the hyperpaths utilized under an optimal sensor deployment strategy, the provision of information leads to the utilization of paths which are not considered under a deterministic routing approach. The rate of increase of the benefits of information, when these are expressed as a fraction of the default system expected cost, is a decreasing function of the system size. This may be used to define optimal fleet sizes for applications in which the system may be subdivided and the number of probes to be utilized is a decision variable.

The IBSO assignment paradigm has many potential applications particularly if we consider the multiple problem variation defined in Chapter 2. The deployment of fleets of emergency vehicles on post-disaster scenarios was described in the introduction. The models may also be used to assist the design of bus routes, which are already used as probes in some cities Chakroborty and Kikuchi [2004], Cathey and Dailey [2002], Dailey and Cathey [2006], Tantiyanugulchai and Bertini [2003]) . Even though there are a multitude of factors determining transit routes, the proposed models may be used to select among a pre-selected set of alternate paths. The same principle may be applied in other contexts, such as the routing of delivery trucks and even taxi cabs. Most of these applications require incorporating one or more of the extensions proposed in Section 7.4. This is likely to increment the complexity of the solution procedure, but it is a promising step towards a more effective utilization of available resources.

7.3 Solution methodologies

The methodological approach used to solve all the numerical examples presented in this dissertation was developed for the solution of the proposed optimal sensor deployment problem. In order to implement the technique to the analysis of IBSO assignment problems, the paths followed by the assets utilized as probes under an IBSO deployment were regarded as sets of sensors deployed on consecutive links. Such approach requires to enumerating all acyclic paths connecting the analyzed origin-destination pair. The later was accomplished implementing a customized path domination criterion which reduced the number of paths to consider by up 74%.

The model formulations are combinatorial in nature, and exact solution approaches are not likely to be effective for large networks. Section 4.5 suggests some mathematical programming approaches, including Benders decomposition and quadratic programming techniques, which may be applicable to the solution of this problem and deserve further consideration. This work implements a methodology based on network optimization methods, taking advantage of the simplifying model assumptions. Such approach lends itself to heuristic implementation, and can easily incorporate changes to the problem assumptions and formulations.

The solution method is based on the fact that, in virtue of the assumptions presented in Chapter 2, the models may be solved by enumerating all feasible sensor/probe deployment strategies, and computing the corresponding expected costs under information. Such approach poses two main challenges: the potentially huge number of perceived states which need to be considered during the evaluation of a feasible deployment strategy, and the existence of a combinatorial number of such strategies. The proposed solution technique deals with the first issue using state-partitioning principles, while the combinatorial aspect of the problem is addressed heuristically, by implementing an adaptive memory Tabu search procedure.

The state-space partitioning algorithm, introduced in Section 5.1, is guided by rules developed specifically for the problems under study. These are used to reduce the number of shortest path computations required to find an optimal solution, mostly by appropriately selecting threshold values for the corresponding cost. Numerical experiments suggest that, in well connected networks, the algorithm may reduce the computational effort by up to 95%. The adaptive memory Tabu search procedure, presented in Section 5.2 explores the combinatorial solution space guided by short and long term memory structures. In the examples studied in Section 5.2.2.2 it found the optimal solution by evaluating between 3% and 20% of all the candidate solutions.

The performance of the combined methodology is very satisfactory, and the results suggest that the heuristic efficiency, in terms of percentage of evaluated strategies is not affected by the network size or the number of deployed sensors. However, the state partitioning technique may not be sufficient to deal with cases involving a very large number of sensors, particularly if they cover most of the paths connecting the analyzed origin-destination pair. The later complicates the identification of threshold values, reducing the effectiveness of the methodology. A possible approach to overcome this problem is the design of more complex partitioning rules which further reduce the number of required evaluations, or/and to implement shortest path re-optimization methods (reviewed in Appendix B) may be implemented to improve the performance of the technique. Eventually, the problem size may require the utilization of heuristic methods. Possible approaches to their development include Monte Carlo sampling (e.g. Alexopoulos [1997]), and the utilization of surrogate values for the threshold shortest path value, which may allow the computation of error bounds. Finally, it is important to notice that the solution of IBSO assignment problems may benefit from approaches leading to an implicit path-enumeration, which is likely to be the subject of further extensions.

7.4 Extensions and future research directions

The problem variations analyzed in this work involve a single origin destination pair, flow-independent link costs, time-invariant link-cost probability distributions, and the a-priori selection of regular assets routes on the IBSO assignment problem. These assumptions allowed for relatively simple model formulations, which were very useful to better understand problem characteristics and behavior. However, practical implementations are likely to benefit from more complex approaches. The following table (Table 7.1) lists some of the more desirable extensions. Notice that the efficient solution of the suggested extended problems may require major methodological changes.

The consideration of multiple origin-destination pairs is a relatively easy modeling extension which would allow considering more general routing cases. It is not expected to greatly complicate the solution procedure, given that the one-to-all and all-to-all shortest path problem variations may also be solved efficiently (Ahuja et al. [2002]). The partitioning rules proposed in this work are applicable to the extended case, but adjustments may be desirable in the search for efficiency. Some preliminary tests on this problem version suggests that as the number of OD pairs becomes closer to the total number of possible pairs, the optimal assignment strategies resemble the maximum-variance based deployments.

Flow dependant link costs are a requirement if the models are applied to

Network	• OD Pairs • Time-dependent link costs • Flow-dependent link costs
Objective Function	• Include measure of variance • Minimize total deployment time
Regular Assets Deployment Strategy	• Time-dependant shortest path • Shortest path with recourse

Table 7.1: Possible problem extensions

design traffic routing strategies, even though they may not be necessary for other problem applications. The incorporation of a time dimension may lead to more realistic models for traffic-related implementations, and is necessary under some of the alternative objective functions and probe deployment strategies proposed below.

The explicit incorporation of a robustness component in the objective function may be valuable for applications very sensitive to the experienced time. Notice that the variance of the solution may also be limited by incorporating constraints into the maximum path length. The formulation may be fully oriented to minimize the total deployment time, including that of assets and probes. This would lead to a very restricted problem under the assumptions considered in this work, but may provide interesting results if sequential probe deployment strategies are considered.

Finally, the strategy used for the deployment of probes on IBSO assignment problems may follow any of the variations suggested in Chapter 2, and the routing paradigm selected for both, regular assets and probes, may be allowed to be adaptive based on self-collected information or/and time dependant.

The models discussed in this dissertation provide a flexible tool to measure and understand the benefits of information in the context of system optimum assignment under information. Based on our findings, and considering the possible extensions described in this section, a number of practical implementations can be devised, which may contribute towards a more effective utilization of information in the optimization of transportation systems and related areas.

Appendices

Appendix A

Additional Data and Results for Chapter 4

Link	ε_1^j	p_1^j	ε_2^j	p_2^j	ε_3^j	p_3^j	ε_4^j	p_4^j
1	16	*0.6*	25	*0.3*	36	*0.1*		
2	21	*0.5*	24	*0.2*	25	*0.2*	39	*0.1*
3	11	*0.4*	13	*0.4*	26	*0.2*		
4	11	*0.7*	30	*0.3*				
5	13	*0.6*	37	*0.2*	39	*0.2*		
6	24	*0.5*	28	*0.3*	31	*0.2*		
7	11	*0.6*	20	*0.3*	24	*0.1*		
8	23	*0.4*	30	*0.3*	34	*0.3*		
9	14	*0.5*	23	*0.4*	34	*0.1*		
10	22	*0.7*	30	*0.3*				
11	35	*0.6*	40	*0.4*				
12	16	*0.5*	25	*0.4*	37	*0.1*		
13	15	*0.3*	17	*0.3*	19	*0.3*	26	*0.1*
14	27	*0.4*	33	*0.3*	40	*0.3*		
15	28	*0.4*	35	*0.3*	37	*0.2*	40	*0.1*
16	25	*0.7*	32	*0.3*				
17	18	*0.7*	24	*0.3*				
18	18	*0.5*	25	*0.3*	29	*0.2*		
19	11	*0.5*	31	*0.4*	37	*0.1*		
20	21	*0.5*	23	*0.5*				
21	12	*0.5*	23	*0.3*	31	*0.2*		

Table A.1: Network 2: Link cost probability distribution (a)

Link	ε_1^j	p_1^j	ε_2^j	p_2^j	ε_3^j	p_3^j	ε_4^j	p_4^j
22	12	*0.3*	15	*0.3*	22	*0.2*	24	*0.2*
23	19	*0.6*	23	*0.2*	37	*0.2*		
24	13	*0.4*	23	*0.3*	34	*0.3*		
25	14	*0.6*	34	*0.2*	39	*0.2*		
26	13	*0.8*	31	*0.1*	32	*0.1*		
27	14	*0.3*	15	*0.3*	27	*0.2*	32	*0.2*
28	10	*0.6*	17	*0.3*	20	*0.1*		
29	16	*0.3*	18	*0.3*	36	*0.2*	39	*0.2*
30	19	*0.4*	24	*0.3*	29	*0.3*		
31	12	*0.4*	13	*0.3*	25	*0.2*	32	*0.1*
32	15	*0.4*	19	*0.3*	25	*0.3*		
33	14	*0.3*	20	*0.3*	25	*0.2*	32	*0.2*
34	23	*0.9*	34	*0.1*				
35	18	*0.3*	19	*0.3*	20	*0.3*	33	*0.1*
36	10	*0.5*	19	*0.4*	39	*0.1*		
37	13	*0.6*	31	*0.3*	35	*0.1*		
38	15	*0.5*	36	*0.3*	39	*0.2*		
39	16	*0.7*	22	*0.3*				
40	10	*0.3*	13	*0.3*	18	*0.3*	34	*0.1*
41	12	*0.9*	31	*0.1*				
42	14	*0.5*	19	*0.3*	32	*0.2*		

Table A.2: Network 2: Link cost probability distribution (b)

Link	ε_1^j	p_1^j	ε_2^j	p_2^j	ε_3^j	p_3^j	ε_4^j	p_4^j	ε_5^j	p_5^j
1	70	*0.33*	73	*0.33*	94	*0.34*				
2	25	*0.33*	35	*0.33*	82	*0.34*				
3	42	*0.33*	48	*0.33*	61	*0.34*				
4	26	*0.2*	31	*0.2*	55	*0.2*	88	*0.2*	90	*0.2*
5	58	*0.33*	70	*0.33*	95	*0.34*				
6	15	*0.5*	73	*0.5*						
7	65	*0.33*	74	*0.33*	75	*0.34*				
8	59	*0.33*	72	*0.33*	98	*0.34*				
9	21	*0.25*	32	*0.25*	85	*0.25*	98	*0.25*		
10	89	*0.5*	96	*0.5*						
11	32	*0.33*	48	*0.33*	67	*0.34*				
12	63	*0.5*	99	*0.5*						
13	66	*0.33*	85	*0.33*	98	*0.34*				
14	6	*0.25*	15	*0.25*	39	*0.25*	58	*0.25*		
15	2	*0.5*	48	*0.5*						
16	61	*0.33*	63	*0.33*	85	*0.34*				
17	16	*0.2*	18	*0.2*	40	*0.2*	52	*0.2*		
18	3	*0.33*	30	*0.33*	50	*0.34*				
19	16	*0.33*	34	*0.33*	71	*0.34*				
20	90	*0.5*	96	*0.5*						
21	21	*0.33*	46	*0.33*	85	*0.34*				
22	17	*0.25*	49	*0.25*	53	*0.25*	65	*0.25*		
23	6	*0.25*	12	*0.25*	54	*0.25*	66	*0.25*		

Table A.3: Network I: Uniform link cost probability distribution

Link	ε_1^j	p_1^j	ε_2^j	p_2^j	ε_3^j	p_3^j	ε_4^j	p_4^j	ε_5^j	p_5^j
1	70	*0.15*	73	*0.15*	94	*0.7*				
2	25	*0.15*	35	*0.15*	82	*0.7*				
3	42	*0.15*	48	*0.15*	61	*0.7*				
4	26	*0.075*	31	*0.075*	55	*0.075*	88	*0.075*	90	*0.7*
5	58	*0.15*	70	*0.15*	95	*0.7*				
6	15	*0.3*	73	*0.7*						
7	65	*0.15*	74	*0.15*	75	*0.7*				
8	59	*0.15*	72	*0.15*	98	*0.7*				
9	21	*0.1*	32	*0.1*	85	*0.1*	98	*0.7*		
10	89	*0.3*	96	*0.7*						
11	32	*0.15*	48	*0.33*	67	*0.7*				
12	63	*0.3*	99	*0.7*						
13	66	*0.15*	85	*0.15*	98	*0.7*				
14	6	*0.1*	15	*0.1*	39	*0.1*	58	*0.7*		
15	2	*0.3*	48	*0.7*						
16	61	*0.15*	63	*0.15*	85	*0.7*				
17	16	*0.1*	18	*0.1*	40	*0.1*	52	*0.7*		
18	3	*0.15*	30	*0.15*	50	*0.7*				
19	16	*0.15*	34	*0.15*	71	*0.7*				
20	90	*0.3*	96	*0.7*						
21	21	*0.15*	46	*0.15*	85	*0.7*				
22	17	*0.1*	49	*0.1*	53	*0.1*	65	*0.7*		
23	6	*0.1*	12	*0.1*	54	*0.1*	66	*0.7*		

Table A.4: Network I: Right-skewed link cost probability distribution

Link	ε_1^j	p_1^j	ε_2^j	p_2^j	ε_3^j	p_3^j	ε_4^j	p_4^j	ε_5^j	p_5^j
1	70	*0.7*	73	*0.15*	94	*0.15*				
2	25	*0.7*	35	*0.15*	82	*0.15*				
3	42	*0.7*	48	*0.15*	61	*0.15*				
4	26	*0.7*	31	*0.075*	55	*0.075*	88	*0.075*	90	*0.075*
5	58	*0.7*	70	*0.15*	95	*0.15*				
6	15	*0.7*	73	*0.3*						
7	65	*0.7*	74	*0.15*	75	*0.15*				
8	59	*0.7*	72	*0.15*	98	*0.15*				
9	21	*0.7*	32	*0.1*	85	*0.1*	98	*0.1*		
10	89	*0.7*	96	*0.3*						
11	32	*0.7*	48	*0.15*	67	*0.15*				
12	63	*0.7*	99	*0.3*						
13	66	*0.7*	85	*0.15*	98	*0.15*				
14	6	*0.7*	15	*0.1*	39	*0.1*	58	*0.1*		
15	2	*0.7*	48	*0.3*						
16	61	*0.7*	63	*0.15*	85	*0.15*				
17	16	*0.7*	18	*0.1*	40	*0.1*	52	*0.1*		
18	3	*0.7*	30	*0.15*	50	*0.15*				
19	16	*0.7*	34	*0.15*	71	*0.15*				
20	90	*0.7*	96	*0.3*						
21	21	*0.7*	46	*0.15*	85	*0.15*				
22	17	*0.7*	49	*0.1*	53	*0.1*	65	*0.1*		
23	6	*0.7*	12	*0.1*	54	*0.1*	66	*0.1*		

Table A.5: Network I: Left-skewed link cost probability distribution

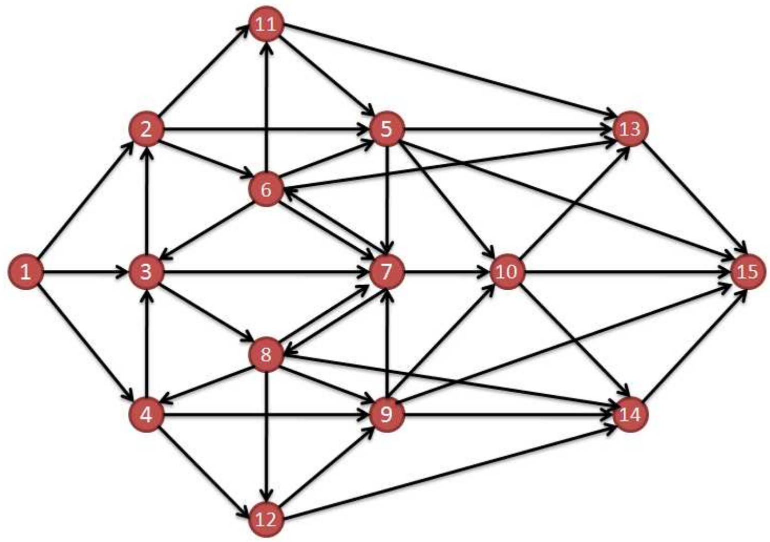

Figure A.1: Network II topology

Link	ε_1^j	p_1^j	ε_2^j	p_2^j	**Link**	ε_1^j	p_1^j	ε_2^j	p_2^j
1	70	*0.5*	94	*0.5*	**13**	66	*0.5*	98	*0.5*
2	25	*0.5*	82	*0.5*	**14**	6	*0.5*	58	*0.5*
3	42	*0.5*	61	*0.5*	**15**	2	*0.5*	48	*0.5*
4	26	*0.5*	90	*0.5*	**16**	61	*0.5*	85	*0.5*
5	58	*0.5*	95	*0.5*	**17**	16	*0.5*	52	*0.5*
6	15	*0.5*	73	*0.5*	**18**	3	*0.5*	50	*0.5*
7	65	*0.5*	75	*0.5*	**19**	16	*0.5*	71	*0.5*
8	59	*0.5*	98	*0.5*	**20**	90	*0.5*	96	*0.5*
9	21	*0.5*	98	*0.5*	**21**	21	*0.5*	85	*0.5*
10	89	*0.5*	96	*0.5*	**22**	17	*0.5*	65	*0.5*
11	32	*0.5*	67	*0.5*	**23**	6	*0.5*	66	*0.5*
12	63	*0.5*	99	*0.5*					

Table A.6: Network I: Two-states link cost probability distribution

Link	ε_1^j	p_1^j	ε_2^j	p_2^j	ε_3^j	p_3^j	ε_4^j	p_4^j	ε_5^j	p_5^j
1	65	*0.2*	70	*0.2*	73	*0.2*	94	*0.2*	99	*0.2*
2	20	*0.2*	25	*0.2*	35	*0.2*	82	*0.2*	86	*0.2*
3	38	*0.2*	42	*0.2*	48	*0.2*	61	*0.2*	65	*0.2*
4	28	*0.2*	31	*0.2*	55	*0.2*	88	*0.2*	90	*0.2*
5	55	*0.2*	58	*0.2*	70	*0.2*	95	*0.2*	99	*0.2*
6	10	*0.2*	12	*0.2*	15	*0.2*	73	*0.2*	78	*0.2*
7	60	*0.2*	65	*0.2*	74	*0.2*	75	*0.2*	78	*0.2*
8	55	*0.2*	59	*0.2*	72	*0.2*	98	*0.2*	99	*0.2*
9	18	*0.2*	21	*0.2*	32	*0.2*	85	*0.2*	98	*0.2*
10	78	*0.2*	85	*0.2*	89	*0.2*	96	*0.2*	98	*0.2*
11	8	*0.2*	32	*0.2*	48	*0.2*	67	*0.2*	69	*0.2*
12	55	*0.2*	60	*0.2*	63	*0.2*	99	*0.2*	100	*0.2*
13	63	*0.2*	66	*0.2*	85	*0.2*	98	*0.2*	99	*0.2*
14	3	*0.2*	6	*0.2*	15	*0.2*	39	*0.2*	58	*0.2*
15	0	*0.2*	1	*0.2*	2	*0.2*	48	*0.2*	52	*0.2*
16	55	*0.2*	61	*0.2*	63	*0.2*	85	*0.2*	89	*0.2*
17	13	*0.2*	16	*0.2*	18	*0.2*	40	*0.2*	52	*0.2*
18	1	*0.2*	3	*0.2*	30	*0.2*	50	*0.2*	55	*0.2*
19	12	*0.2*	16	*0.2*	34	*0.2*	71	*0.2*	75	*0.2*
20	80	*0.2*	85	*0.2*	90	*0.2*	96	*0.2*	99	*0.2*
21	18	*0.2*	21	*0.2*	46	*0.2*	85	*0.2*	88	*0.2*
22	15	*0.2*	17	*0.2*	49	*0.2*	53	*0.2*	65	*0.2*
23	3	*0.2*	6	*0.2*	12	*0.2*	54	*0.2*	66	*0.2*

Table A.7: Network I: Five-states link cost probability distribution

Appendix B

Shortest Path Re-Optimization

The exact solution of the problem described in this Chapter 4 involves finding a large number of shortest paths between every origin/destination pair in the network, corresponding to all perceived scenarios. The only difference among such scenarios is the cost on the instrumented links, and therefore the model may benefit from the implementation of shortest path re-optimization techniques.

Short es path re-optimization methodologies find the new shortest path on a network after the cost of one or more arcs changes, based on previous shortest path computations. Under some conditions, these techniques are theoretically more efficient than recomputing the shortest path from scratch. Moreover, speedups of up to five orders of magnitude have been found in practical experiments conducted no randomly generated networks (Demetrescu et al. [2004]) and on real transportation systems(Demetrescu et al. [2004],Buriol et al. [2003]) .

Several algorithms are available in the literature, appropriate for different hypotheses regarding the number of links which cost changes simultaneously, the direction of such adjustments, and the admissible values for the link weights. The first efforts (Dionne [1978], Rodionov [1968], Murchland [1970], Goto and Sangiovanni-Vincentelli [1978], Frigioni et al. [2000]) proposed methodologies that may be used to update the shortest path tree when a single arc cost is either incremented or reduced. Other authors (Gallo [1980], Gallo and Pallottino [1982], Nguyen et al. [2002]) present efficient solution techniques for problems in which the cost in exactly one link is reduced, which are also appropriate for situations in which the root node of the shortest path is modified. Fujishige [1981] presents an algorithm to analyze the case of a cost reduction on a set of arcs incident to a common node. More recent approaches (Ramalingam and Reps [1996], Pallotino and Scutella [2003], King [1999]) are fully dynamic, in the sense that can be utilized to analyze cost reductions and/or increments on any subset of arcs, as well as arc insertions and/or deletions. These approaches, which are suitable for the solution of the optimal sensor deployment problem, are briefly discussed in the following paragraphs. Recent work by Miller-Hooks and Yang [2005], which extends general shortest path re-optimization algorithms to the case of time-varying networks, escapes the scope of this thesis and it is not discussed.

King [1999] proposes a fully dynamic re-optimization algorithm which works on graphs with small integer weights. The methodology maintains a pair of shortest paths of length $\leq d$going in and out of each node, and "stitches" them together in order to obtain shortest paths of larger length. By choosing an appropriate value of d, the algorithm can perform updates in $O(n^{2.5}\sqrt{C\,log\,n})$, where C is the maximum arc weight. The required storage space is in the order of $O(n^{2.5}\sqrt{C})$.

Ramalingam and Reps [1996] present one of the most popular algorithms to solve the one-to-all shortest path re optimization problem in networks with strictly positive real-valued arc weights. Their approach has a worst case complexity of $O(m_a + n_a\,log\,n_a)$, where $m_a$and n_a are set of the arcs and nodes affected by the costs changes, respectively.

C.Demetrescu and Italiano [2001] present a fully dynamic shortest path algorithm designed to maintain all-to-all shortest paths in directed networks with real-valued edge weights. Their methodologies depart from a matrix viewpoint of the shortest path problem, in virtue of which the optimal distances on a directed graph can be obtained by performing specific operations on the weights matrix. The proposed algorithms have a better worst case complexity than recomputing the shortest path from the beginning and are able to accommodate cost changes of any magnitude.

In a later paper Demetrescu and Italiano [2003] propose a new approach, valid for directed graphs with non negative arc weights. The algorithm relies on efficiently maintaining a sets of paths with specific properties, from which the shortest paths can be obtained in $O(1)$. The authors introduce the concept of potentially uniform paths, which consist of proper sub paths that are either shortest paths or historical shortest paths. The later are paths which were the shortest before a network update, and whose arcs were not affected by the corresponding changes. By bounding the maximum number of new potentially uniform paths after each network update in a sequence, the authors are able to compute the worst case complexity of their procedure, which is in the order of $O(n^2 \, log \, n)$. This is achieved by reducing the number of historical shortest paths generated after each network modification, by appropriately "smoothing" the sequence of changes which conform a network update.

Pallotino and Scutella [2003] present a framework for the one to all shortest path re optimization problem on networks with integer-valued link weights based on the reduced cost of the network arcs. Reduced costs are obtained from a linear programming formulation of the shortest path problem as $\bar{c_{ij}} = \pi_j - \pi_i - c_{ij}$, where c_{ij} is the non negative cost associated with arc ij, and π_i is the optimal node potential of node i, equal in value to the dual variable associated to node i. When an optimal solution to the problem is found, the reduced costs must satisfy the feasibility and optimality conditions displayed in B.1 and B.2 respectively, where T_r^* denotes the shortest path tree rooted at node r.

$$\bar{c_{ij}} \geq 0 \forall ij \in A \tag{B.1}$$

$$\bar{c_{ij}} = 0 \forall ij \in T_r^* \tag{B.2}$$

When the cost on one or more arcs changes, T_r^* may no longer be optimal, which translates into violated optimality and/or feasibility conditions. The algorithm proposed by Pallotino and Scutella [2003] deals separately with such violations. In a first stage, it updates the tree with respect to the arcs in $K^+ = \{ij : \bar{c_{ij}} > 0, ij \in T_r^*\}$ by means of a dual-based tree-hanging procedure previously presented by Pallottino and Scutella [1997]. The origin-based sub tree obtained by removing the tree arcs with $\bar{c_{ij}} > 0$, which is clearly part of the updated shortest path, is progressively extended by appropriately increasing the potential of the remaining nodes in such way the complementary slackness conditions are met by the updated tree arcs, while maintaining dual feasibility. The main contribution of this work is the analysis of a set of properties in virtue of which entire sub trees can be added to the updated tree in a single step. Furthermore, it provides the conditions under which several sub trees can be incorporated simultaneously, which improves the practical performance of the algorithm with respect to previous dual ascent procedures (Florian et al. [1981], Gallo and Pallottino [1982]). The running time of the dual phase is never worse than the best strongly polynomial implementation of Dijkstra's shortest path algorithm (Dijkstra [1959]) $O(m + n\,log\,n)$. Furthermore, the complexity can be also bounded based on the size of the initial sub tree n_r, and on the maximum path cost change after the perturbations, C_d.

The second stage takes as an input a reduced graph G^- consisting on the previously updated tree plus all the arcs in $K^- = \{ij : \bar{c_{ij}} < 0, ij \in T_r^*\}$. The procedure implemented in this step restores feasibility by updating the node potentials in such way that $\bar{c_{ij}} \geq 0\ \forall ij \in A$ after all the cost reductions are implemented. This is accomplished in phases, each of which considers a "star path" sub graph of G^-. The structure of such sub graphs is such that, when the corresponding cost changes are implemented, the node potentials can be updated using a label setting algorithm. The authors provide a methodology to detect a star paths in $O(m)$, which leads to a worst case complexity of for the primal phase of $O(hm + h\,n\,log\,n)$, where the second term represents the complexity of the label setting algorithm, and h is the maximum number of phases, bounded by the magnitude of the maximum cost change.

The conducted review suggests that there are a number of shortest path re optimization algorithms available in the literature which may be used to improve the performance of the methodologies considered in this dissertation.

Appendix C

Computing Paths Properties on Stochastic Networks

When the inherent uncertainty regarding network parameters is explicitly considered, the difficulty of finding network properties, and even of solving the simplest optimization problems, may grow exponentially. The problems analyzed in this dissertation implicitly involve the identification of shortest paths properties on stochastic networks, including the probability of a path being the shortest, and the probabilities that a link belongs to the shortest path under information. The following paragraphs provide a brief overview of previous studies which illustrate the challenges involved in such computations.

Finding the Least Expected Cost (LEC) path is a relatively easy task when the link cost functions are linear. In virtue of the definition of the expected cost of a sum of random variables, it is valid to replace arc probability distributions by their expected costs, and run a deterministic shortest path algorithm on the resulting network (Eiger et al. [1985]).

The computation of path properties on stochastic networks has been proved to be substantially more difficult that the identification of the LEC path. The main reason for this is that computing stochastic path properties usually entails generating the corresponding Probability Distribution Functions (PDFs) based on links PDFs. This can easily become mathematically intractable in networks with more than a few links, particularly when links

PDFs are either continuous or discrete with a large support. For this reason, most of the papers on this field present heuristics to approximate the desired properties.

Frank [1968] analyzes the PDF of the shortest path length on a network with continuous arc costs. He provides a closed-form optimal solution entailing the computation of an r-dimensional integral, were r can be as large as the total number of links on the network. He also proposes an approach to perform non parametrical analysis based on random sampling, and implements a Monte Carlo simulation scheme to show that the shortest path length PDF is approximately normal when the link costs are uniform, normal or exponential.

Sigal et al. [1980] study the problem of finding the probability of a path being shorter than every other path. They derive closed form solutions for the case of continuous link cost distribution functions, and present a cutset solution approach which may entail enumerating all possible cuts. The authors suggest several heuristic techniques to solve the proposed problem.

Using a framework closer to the one adopted for this work, Alexopoulos [1997] focuses his research on shortest path and minimum spanning tree problems on networks with stochastic arc costs described by independent discrete probability distribution functions. He shows that computing properties such as the expected length of the shortest path, the probability of a path being the shortest, and the probability of an arc belonging to a shortest path, are #P-hard problems, and proposes a state partitioning approach to approximate the solution. The underlying concept of his approach is that the desired path properties can be computed without explicitly considering all network states. Based on the assumption that the link states are sorted in an increasing order of costs, it is possible to define boundaries, limiting the combinations of link states which need to be considered for the computation of a particular property. The procedure is exponential in the worst case, but some practical applications suggest a better performance. Furthermore, Monte Carlo simulation may be implemented to efficiently approximate the problem solution in large networks.

Kim et al. [2005] use a state space reduction approach to identify those links which do not need to be considered during shortest path computations. They study online vehicle routing problems on non-stationary stochastic networks with two states per arc, which exhibit Markovian dependencies across time intervals. The authors provide algorithms for both, the a-priori elimination of irrelevant arcs, and the dynamic identifications of such arcs as the vehicle moves through the network.

This brief review suggests that, even though the computation of the properties required to evaluate sensor deployment strategies for IBSO assignment is challenging, there is a wide variety of promising approaches which can allow for a relatively efficient solution of the problem.

Bibliography

W.P. Adams and H.D. Sherali. A tight linearization and an algorithm for zero-one quadratic programming problems. *Management Science*, 32,(10): 1274–1290, 1986.

J.L. Adler and M.G. McNally. In-laboratory experiments to investigate driver behavior under advanced traveler information systems. *Transportation Research Part C: Emerging Technologies*, 2(3):149–164, 1994.

A.Ehlert, M.G.H. Bell, and S.Grosso. The optimisation of traffic count locations in road networks. *Transportation Research Part B*, 40:460–479, 2006.

A.Faye and F. Roupin. Partial lagrangian relaxation for general quadratic programming. *4OR: A Quarterly Journal of Operations Research*, 5(1):75–88, 2007.

R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows*. Prentice Hall, 1993.

R.K. Ahuja, O.Ergunb, J. B. Orlin, and A.P. Punnend. A survey of very large scale neighborhood search techniques. *Discrete Applied Mathematics*, 123: 75–102, 2002.

C. Alexopoulos. State space partitioning methods for stochastic shortest path problems. *Networks*, 30:9–21, 1997.

G. Andreatta and L. Romeo. Stochastic shortest paths with recourse. *Networks*, 18:193–204, 1988.

R. Arnott. Does providing information to drivers reduce traffic congestion? *Transportation Research A*, 25:309–318, 1991.

A. Balakrishnan, T. Magnati, and R.T. Wong. A dual ascent procedure for large-scale uncapapcitated network design. *Operations Research*, 37(5):717–740, 1989.

J.L. Bander and C.C White. A heuristic search approach for a nonstationary stochastic shortest path problem with terminal costs. *Transportation Science*, 36(2):218–230, 2002.

H. Bar-Gera. Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times: A case study from israel. *Transportation Research Part C: Emerging Technologies*, 15(6):380–391, 2007.

M. Ben-Akiva, A. De Palma, and I. Kaysi. Dynamic network models and driver information systems. *Transportation Research*, 25(5):251–266, 1991.

J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4, 1962.

L. Bianco, G. Confessore, and P. Reverberi. A network based model for traffic sensor location with implications on o/d matrix estimates. *Transportation Science*, 35(1):50–60, 2001.

K. Biesecker. Broadband wireless, integrated services, and their application to intelligent transportation systems. Technical report, Federal Highway Administration, 2000.

E. Boros and P. L. Hammer. Pseudo boolean optimization. *Discrete Applied Mathematics*, 123:155–225, 2002.

D.R. Boyce, B. Ran, and L. Leblanc. Solving an instantaneous dynamic user-optimal route choice model. *transportation Science*, 29(2):128–142, 1995.

L. S. Buriol, M.G.C. Resende, and Mikkel Thorup. Speeding up dynamic shortest path algorithms. Technical report, AT&T Labs Research, 2003.

R.E. Burkard and T. Bonniger. A heuristic for quadratic boolean programs with applications to quadratic assignment problems. *European Journal of Operational Research*, 13(4):374–386, 1983.

F.W Cathey and D. J. Dailey. Transit vehicles as traffic probe sensors. *Transportation Research Record*, 1804:23–30, 2002.

R. Cayford and T. Johnson. Operational parameters affecting the use of anonymous cell phone tracking for generating traffic information. In *Annual Meeting of the Transportation Research Board*, Washington, D.C., 2003.

R. Cayford and Y.B. Yim. A field operation test using anonymous cell phone tracking for generating traffic information. In *Transportation Research Board 2006 annual meeting CD-ROM paper number 2511*, 2006.

C.Demetrescu and G.F. Italiano. Fully dynamic all pairs shortest paths with real edge weights. In *IEEE Symposium on Foundations of Computer Science*, pages 260–267, 2001.

P. Chakroborty and S. Kikuchi. Using bus travel time data to estimate travel times on urban corridors. *Transportation Research Record*, 1870:18–25, 2004.

J. L. Charbonnier, J.L. Farges, and J.J. Henry. Models and strategies for dynamic route guidance-part c: Optimal control approach. In *DRIVE Conference*, pages 106–112, Brussels, Belgium, 1991.

Y.Y. Chen, M.G.H. Bell, and K. Bogenberger. Reliable pre-trip multi-path planning and dynamic adaptation for a centralized road navigation system. In *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*. IEEE, September 2005.

R.K. Congram, C.N. Potts, and S.L. van de Velde. An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem. *Informs Journal of Computing*, 14(1):52–67, 2002.

A.M. Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Comput. Oper. Res.*, 32(6):1429–1450, 2005.

D.J. Dailey and F.W. Cathey. Deployment of a virtual sensor system, based on transit probes, in an operational traffic management system. Technical report, Washington State Transportation Center (TRAC), 2006.

J.J. Davies, A.R. Beresford, and A. Hopper. Scalable, distributed, real-time map generation. *Pervasive Computing, IEEE*, 5(4):47–54, 2006.

R. De Leone and D. Pretolani. Auction algorithms for shortest hyperpath problems. Technical report, Universita de Camerino, 1998.

C. Demetrescu and G. Italiano. A new approach to dynamic all pairs shortest paths. In *35th Annual ACM Symposium on Theory of Computing*, pages 159–166, San Diego, CA, 2003.

C. Demetrescu, S. Emiliozzi, and G. Italiano. Experimental analysis of dynamic all pairs shortest path algorithms. In *15th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2004.

E. W. Dijkstra. A note on two problems in connexion with graphs. *Numer. Math.*, 1:269–271, 1959.

R. Dionne. Étude et extension de un algorithme de murchland. *INFOR*, 16: 132–146, 1978.

A. Eiger, P. Mirchandani, and H. Soroush. Path preferences and optimal paths in probabilistic networks. *Transportation Science*, 19(1):75–84, 1985.

X. Fei, H.S. Mahmassani, and S.M. Eisenman. Sensor coverage and location for real-time traffic prediction in large-scale networks. *Transportation Research Record: Journal of the Transportation Research Board*, 2039:1–15, 2007.

F.Glover. Tabu search. part 1. *ORSA Journal on Computing*, 1(3):190–206, 1989.

F.Glover. Tabu search. part 2. *ORSA Journal on Computing*, 2:4–32, 1990.

M. Florian, S. Nguyen, and S. Pallottino. A dual simplex algorithm for finding all shortest paths. *Networks*, 11(4):367 – 378, 1981.

M.D. Fontaine and B.L. Smith. Probe-based traffic monitoring systems with wireless location technology: an investigation of the relationship between system design and effectiveness. *Transportation Research Record*, 1925:3–11, 2005.

S. Foo, A. Barkan, A. Gravitis, and M. Org. Freeway travel time prediction and route recommendation via cell phone. In *TRB 85th Annual Meeting Compendium of Papers*, 2006.

H. Frank. Shortest paths in probabilistic graphs. *Operations Research*, 17(4): 583–599, 1968.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

T.L. Friesz, J. Luque, R. L. Tobin, and B.W. Wie. Dynamic network traffic assignment considered as a continuous time optimal control problem. *Oper. Res.*, 37(6):893–901, 1989.

D. Frigioni, A. Marchetti-Spaccamela, and U. Nanni. Fully dynamic algorithms for maintaining shortest paths trees. *J. Algorithms*, 34:251–281, 2000.

L. Fu. An adaptive routing algorithm for in-vehicle route guidance systems with real-time information. *Transportation Research Part B: Methodological*, 35(8):749 – 765, 2001.

S. Fujii and R. Kitamura. Anticipated travel time, information acquisition and actual experience: The case of hanshin expressway route closure. *Transportation Research Record*, 1725:79–85, 2000.

S. Fujishige. A note on the problem of updating shortest paths. *Networks*, 11: 317–319, 1981.

G. Gallo. Reoptimization procedures in shortetst path problems. *Decisions in Economics and Finance*, 3(1):3–13, 1980.

G. Gallo and S. Pallottino. A new algorithm to find the shortest paths between all pairs of nodes. *Discrete Appl. Math*, 4:23–35, 1982.

S. Gao. *Optimal adaptive routing and traffic assignment in stochastic time dependent networks*. PhD thesis, M.I.T, 2005.

S. Gao and I Chabini. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B*, 40:93–122, 2006.

A.M Geoffrion and G. Graves. Multicomodity distribution system design by bender's decomposition". *Management science*, 5:822–844, 1974.

F Glover. Heuristics for integer programming using surrogate constraints. *Decision Sciences*, 8:156–166, 1977.

F. Glover, G.A. Kochenberger, and B. Alidaee. Adaptive memory tabu search for binary quadratic programs. *Management Science*, 44(3):336–345, 1998.

S. Goto and A. Sangiovanni-Vincentelli. A new shortest path updating algorithm. *Networks*, 8:341–372, 1978.

G.J. Gutierrez, P. Kouvelis, and A.A. Kurawarwala. A robustness approach to uncapacitated network design problems. *European Journal of Operational Research*, 94(2):362–376, 1996.

R. Hall. The fastest path through a network with random time-dependent travel times. *Transp. Sci.*, 20:182–188, 1986.

Y. Hamdouch, P. Marcotte, and S. Nguyen. A strategic model for dynamic traffic assignment. *Networks and Spatial Economics*, 4:291–315, 2004.

J. K. Ho. A successive linear optimization approach to the dynamic traffic assignment problem. *Transportation Science*, 14(4):295–305, 1980.

K. Holmberg and J. Hellstrand. Solving the uncapacitated network design problem by a lagrangean heuristic and branch-and-bound. *Oper. Res.*, 46 (2):247–259, 1998.

O. Jahn, R.H. Möhring, A.S. Schulz, and N.E. Stier-Moses. System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations Research*, 53(4):600–616, 2005.

R. Jayakrishnan, H.S. Mahmassani, and T.Y. Hu. An evaluation tool for advanced traffic information and management systems in urban networks. *Transportation Research Part C: Emerging Technologies*, 2(3):129 – 147, 1994.

M. Senouci Jerbi, S.-M. Rasheed, and T. Ghamri-Doudane. An infrastructure-free traffic information system for vehicular networks. In *Vehicular Technology Conference*, 2007.

J. Jin, Z. Qiu, and B. Ran. Effect analysis on parameters of route travel time detection method using network-based wireless location technologies. In *TRB 86th Annual Meeting*, 2007.

W-L. Jin and W. W. Recker. Instantaneous information propagation in a traffic stream through inter-vehicle communication. *Transportation Research Part B: Methodological*, 40(3):230–250, 2006.

D.E. Kaufman, J. Nonis, and R.L. Smith. A mixed integer linear programming model for dynamic route guidance. *Transportation Research Part B: Methodological*, 32(6):431 – 440, 1998.

S. Kim, M. E. Lewis, and C.C. White III. State space reduction for nonstationary stochastic shortest path problems with real-time traffic information. *IEEE Transactions on Intellignet Transportation Systems*, 6 (3):273–284, 2005.

Y. B. Y. Kim and Ra.Cayford. Investigation of vehicles as probes using global positioning system and cellular phone tracking. Technical report, University of Californa at Berkeley, 2000.

V. King. Fully dynamic algorithms for maintaining all-pairs shortest paths and transitive closure in digraphs. pages 81–91, 1999.

R. Kitamura and S. Nakayama. Can travel time information really influence network flow? implications of the minority game. In *TRB 86th Annual Meeting*, 2007.

G.A. Klunder and H.N. Post. The shortest path problem on large-scale real-road networks. *Networks*, 48(4):182–194, 2006.

B. Korte and J. Vygen. *Combinatorial Optimization.* Springer, Berlin, Germany, 2000.

H. Koustopoulos, A. Polydopolou, and M. Ben-Akiva. The use of travel simulators to investigate the response to traffic information. Technical report, M.I.T, 1993.

J. Kwon, K. Petty, and P.P. Varaiya. Probe vehicle runs or loop detectors? effect of detector spacing and sample size on accuracy of freeway congestion monitoring. In *TRB 86th Annual Meeting*, Washington DC, 2007.

E.L. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart & Winston, New York, 1976.

C.K. Lee. A multiple-path routing strategy for vehicle route guidance systems. *Transportation Research Part C*, 2:185–195, 1994.

L.Engelson. On co-ordination of traffic information from multiple sources. Technical report, KFB, 2000.

T.L. Magnanti and R. T. Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations research*, 29:464–484, 1981.

T.L. Magnanti, P. Mireault, and R.T. Wong. Tailoring benders decomposition for network design. *Mathematical Programming Study*, 26:112–154, 1986.

H. S. Mahamassani and S. Peeta. Network performance under system optimal and user equilibrium dynamic assignments: Implications for advanced traveller information systems. *Transp. Res. Rec.*, 1408:83–93, 1994.

H. Mahmassani and P. Chen. Comparative assessment of origin-based and en route real-time information under alternative user behavior rules. *Transportation Research Record*, 1306:69–81, 1991.

H. Mahmassani and C.C. Tong. Availability of information and dynamics of departure time choice : Experimental investigation. *Transportation Research Record*, 1085:33–47, 1986.

H. S. Mahmassani and R. Jayakrishnan. System performance and user response under real-time information in a congested traffic corridor. *Transportation Research A*, 25(5):293–307, 1991.

P. Marcotte, S. Nguyen, and A. Schoeb. A strategic flow models of traffic assignment in static capacitated networks. *Operations Research*, 52(2):191–212, 2004.

A. G. Marín and P. Jaramillo. Urban rapid transit network design: accelerated benders decomposition. *Annals of Operations Research*, 169(1):35–53, 2008.

W. M.Carlyle and R. K.Wood. Near-shortest and k-shortest simple paths. *Networks*, 46(2):98–109, 2005.

Deepak K. Merchant and George L. Nemhauser. A model and an algorithm for the dynamic traffic assignment problems. *Transportation Science*, 12(3): 183–199, 1978.

A. Messmer and M. Papageorgiou. Automatic control methods applied to freeway network traffic. *Automatica*, 30:691–702, 1994.

E. Miller-Hooks. Adaptive least expected time paths in stochastic, time-varying transportation and data networks. *Networks*, 37(1):35–52, 2001.

E. Miller-Hooks and H. Mahmassani. Optimal routing of hazardous materials in stochastic, time-varying transportation networks. *Trans Res Rec*, 1645: 143–151, 1998.

E. Miller-Hooks and B. Yang. Updating paths in time-varying networks given arc weight changes. *Trasportation Science*, 39(2):451–464, 2005.

E.D. Miller-Hooks and H.S. Mahmassani. Least expected time paths in stochastic,time-varying transportation networks. *Transportation Science*, 34:198–215, 2000.

J.E. Moore II, S. Cho, A. Basu, and D.B. Mezger. Use of los angeles freeway service patrol vehicles as probe vehicles. Technical report, California PATH, University of Southern California, 2001.

H. Moritsu. Analysis of traffic network flow by adapting route guidance. In *Infrastructure Planning*, volume 9, pages 37–44, 1991.

J. Murchland. *A fixed matrix method for all shortest distances in a directed graph and for the inverse problem*. PhD thesis, University of Karlsruhe, Karlsruhe, Germany, 1970.

S. Nguyen and S. Pallottino. *Hyperpaths and Shortest Hyperpaths*, volume 1403, pages 258–271. Springer-Verlag, Berlin, 1989.

S. Nguyen, S. Pallottino, and M.G. Scutellà. *A new dual algorithm for shortest path reoptimizationTransportation and Network Analysis-Current Trends*, chapter A new dual algorithm for shortest path reoptimization, pages 221–235. Kluwer, Dordrecht, 2002.

J-S. Oh and R. Jayakrishnan. Emergence of private advanced traveler information service providers and its effect on traffic network performance. *ransportation Research Record: Journal of Transportation Research Board*, 1783:167–177, 2002.

S. Pallotino and M.G. Scutella. A new algorithm for reoptimizing shortest paths when the arc costs change. *Operations research letters*, 31(2):149–160, 2003.

S. Pallottino and M.G. Scutella. Dual algorithms for the shortest path tree problem. *Networks*, 29:125–133, 1997.

M. Papageorgiou. Dynamic modelling, assignment and route guidance in traffic networks. *Transp. Res. B*, 24:471–495, 1990.

M. Papageorgiou and A. Messmer. Dynamic network traffic assignment and route guidance via feedback regulation. *Transp. Res. Rec.*, 1306:49–58, 1991.

P. M. Pardalos and G. P. Rodgers. Computational aspects of a branch and bound algorithm for quadratic zero-one programming. *Computing*, 45(2): 131–144, 1990.

Y. Pavlis and M. Papageorgiou. Simple decentralized feedback strategies for route guidance in traffic networks. *Transportation Science*, 33(3):264–278, 1999.

S. Peeta and A. Ziliaskopoulos. Fundamentals of dynamic traffic assignment: the past, the present and the future. *Networks and Spatial Economics*, 1: 201–230, 2002.

E. P.Hart, N. J.Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern*, 4(2): 100–107, 1968.

J. Polak and P. Jones. The acquisition of pre-trip information : A staded preference approach. *Transportation*, 20(2):179–198, 1993.

G.H. Polychronopoulos and J.T. Tsitsiklis. Stochastic shortest path problems with recourse. *Networks 27, 133-143 (1996).*, 27:133–143, 1996.

D. Pretolani. A directed hypergraph model for random time dependent shortest paths. *European Journal of Operational Research*, 123, 1998.

H. Psaraftis and J. Tsitsiklis. Dynamic shortest paths in acyclic networks with markovian arc costs. *Operations Research*, 41:91–101, 1993.

G. Ramalingam and T. Reps. An incremental algorithm for a generalization of the shortest-path problem. *J. Algorithms*, 21:267–305, 1996.

B. Ran, H. Tang, J. Lu, and S. Lu. Locating roadside servers for advanced traveler information systems. In *TRB 85th Annual Meeting*, 2006.

B. Roberts and D.P. Kroese. Estimating the number of s-t paths in a graph. *Journal of Graph Algorithms and Applications*, 11(1):195–214, 2007.

V. Rodionov. The parametric problem of shortest distances. *U.S.S.R. Computational Math. Math. Phys.*, 8:336–343, 1968.

K. H. Rosen. *Discrete Mathematics and Its Applications*. McGraw-Hill, NY, 2nd edition, 1991.

N. Ruiz Juri, A. Unnikrishnan, and S. T. Waller. Integrated traffic simulation-statistical analysis framework for online prediction of freeway travel time. *Transportation Research Record*, 2039:24–31, 2007.

E.J. Schmitt and H. Jula. Vehicle route guidance systems: Classification and comparison. In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 242–247, Sept. 2006.

S. Sen, R. Pillai, S. Joshi, and A. K. Rathi. A mean-variance model for route guidance in advanced traveler information systems. *Transportation Science*, 35(1):37–49, 2001.

Y. Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, Englewood Cliffs, N.J., 1985.

H. Sherali, J. Desai, H. Rakha, and I. El-Shawarby. A discrete optimization approach for locating automatic vehicle identification readers for the provision of roadway travel times. *transportation Research Part B*, 40:857–871, 2006.

H.D. Sherali and W.P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.

H.D. Sherali and B.M.P. Fraticelli. Enhancing rlt relaxations via a new class of semidefinite cuts. *Journal of Global Optimization*, 22:233–261, 2002.

S. Shladover, G. Polatkan, R. Sengupta, J. VanderWerf, M. Ergen, and B. Bougler. Cooperative vehicle system performance dependence on market penetration. In *TRB 86th Annual Meeting*, 2007.

C.E. Sigal, A.B. Pritsker, and J.J. Solberg. The stochastic shortest route problem. *Operations research*, 28:1122–1129, 1980.

V. P. Sisiopiku, N.M. Rouphail, and A. Santiago. Analysis of correlation between arterial travel time and detector data from simulation and field studies. *Transportation Research Record*, 1457:166–173, 1994.

V.P. Sisiopiku. Wireless information transmission services for intelligent transportation systems applications. In *Proceedings of the 10th ITS America Meeting*, Washington DC, 2000.

K. K. Srinivasan and H.S. Mahamassani. Dynamic decision and adjustment processes in commuter behavior under real-time information. Technical Report 167204-1, Southwest Region University Transportation Center, 2002.

S. Tantiyanugulchai and R.L. Bertini. Arterial performance measurement using transit buses as probe vehicles. In *Intelligent Transportation Systems, 2003. Proceedings*, 2003.

D. Teodorovic, M. Van Aerde, F. Zhu, and F. Dion. Genetic algorithms approach to the problem of automated vehicle identification equipment location. *Journal of Advanced Transportation*, 36(1):1–21, 2002.

G. B. Thomas. The relationship between detector location and travel characteristics on arterial streets. *ITE Journal*, 69(10):36–42, 1999.

A. Unnikrishnan. *Equilibrium models accounting for uncertainty and information provision in transportation networks*. PhD thesis, University of Texas at Austin,, 2008.

L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979a.

L.G. Valiant. The complexity of enumeration and reliability problems. *Siam Journal of Computing*, 8(3):410–421, 1979b.

N.J. van der Zijpp and S. F. Catalano. Path enumeration by finding the constrained k-shortest paths. *Transportation Research Part B: Methodological*, 39(6):545 – 563, 2005.

U. Varshney. Mobile and wireless information systems: Applications, networks, and research problems. *Communications of AIS*, 12:155–166, 2003.

U. Varshney. Vehicular mobile commerce: Applications, challenges and research problems. *Communications of the Association for Information systems*, 16:329–339, 2005.

J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ,, 1944.

S.T. Waller and A. K. Ziliaskopoulos. On the online shortest path problem with limited arc cost dependencies. *Networks*, 40(4):216–227, 2002.

S.T. Waller, J.L. Schofer, and A.K. Ziliaskopoulos. Evaluation with traffic assignment under demand uncertainty. *Journal of the Transportation Research Board*, 1771:69–75, 2001.

X. Wang. Modeling the process of information relay through inter-vehicle communication. *Transportation Research Part B*, 41:684–700, 2007.

D. Watling and T. Van Vuren. The modeling of dynamic route guidance systems. *Transportation Research C*, 2:159–182, 1993.

W.L., W. Chuanjiu, S. Xiaorong, and F. Yuezu. Probe vehicle sampling for real-time traffic data collection. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 222–224, Sept. 2005.

B. Yang and E. Miller-Hooks. Determining critical arcs for collecting real-time travel information. *Transportation Research RecordTransportation Research Record*, 1783:34–41, 2002.

F. Yang, H. X. Liu, R. R. He, X. Ban, and B. Ran. Bilevel formulation for optimal traffic-information dissemination. *Transportation Research Record*, 1836:21–28, 2003.

Q. Yang and H.N. Koutsopoulos. A microscopic traffic simulator for evaluation of dynamic traffic management systems. *Transportation Research Part C: Emerging Technologies*, 4(3):113 – 129, 1996.

X. Yang. *Assessment of a self-organizing distributed traffic information system: modeling and simulation*. PhD thesis, University of California, Irvine, 2003.

X. Yang and W. Recker. Simulation studies of information propagation in a self-organizing distributed traffic information system. *Transportation Research Part C: Emerging Technologies*, 13(5-6):370–390, 2005.

T. Yoshii. Impacts of the accuracy of traffic information in dynamic route guidance systems. In *The Third Annual World Congress on Intelligent Transport Systems*, Orlando, 1996.

T. Yoshii and M. Kuwahara. An evaluating method on effects of dynamic traffic information. In *From vision to reality : proceedings of the 7th World Congress on Intelligent Transportation Systems ITS*, Turin, Italy, 2000.

A. K. Ziliaskopoulos and J. Zhang. A zero public infrastructure vehicle based traffic information system. In *Transportation Research Board Meeting*, 2003.

Vita

Natalia Ruiz Juri was born in Cordoba, Argentina on December 25, 1976, the daughter of Pedro Ruiz Cresta and Ana Maria Juri. She graduated from Colegio de Maria High School in 1994. In May 2001 she finished the 6-year long Civil Engineering program at the National University at Cordoba (Argentina). As an undergraduate student, she was a surveying teaching assistant for two years, and served as an intern at the City of Cordoba. During the last semester of the program, she was employed at Marchessini & Rusculleda, a local construction company, where she participated in several architecture and engineering projects. After graduation, she joined the Institute of Transportation Engineering (ISIT) at the National University at Cordoba, where she collaborated in traffic studies and co-authored a paper. On May 2001 she was awarded a Fulbright scholarship to pursue her master's degree in the United States, joining the Transportation Engineering Graduate Program at the University of Texas at Austin in August 2002. She worked as a Graduate Research Assistant, participating in various projects, and published papers on Integrated Transportation and Land-Use Modeling and Travel Time Prediction. She received the degree of Master of Science in Civil Engineering in August 2004 from the University of Texas at Austin, where she continued her doctoral studies.

Permanent Address: 1910 Morcillo St., Cordoba (5014), Argentina.
This manuscript was typed by the author.